



## Firmware Download

### Format Specification

**3486 8968 Revision B**

**April 12, 1996**

Preliminary  
Specification  
Layout

for Upgrade and new  
program for DFASST

APRIL 1996

E 1

**FIRMWARE DOWNLOAD FORMAT SPECIFICATION**

---

**DOCUMENT CONTROL INFORMATION**

CEDB controls the latest revision of this document. Consult Engineering Records of the DCL for assistance.

Engineering signatures appear in the EIR releasing this document.

This document, including the information contained herein, is restricted, confidential and proprietary to Unisys Corporation, and is to be used only by and disclosed only to those within Unisys Corporation with a need-to-know. Release outside Unisys Corporation requires authorization in writing by a level C or higher Executive of Unisys Corporation or by a designee thereof.

---

**FIRMWARE DOWNLOAD FORMAT SPECIFICATION****CONTENTS**

<b>1. SCOPE</b>	<b>5</b>
<b>2. PURPOSE</b>	<b>6</b>
<b>3. PROCEDURE</b>	<b>7</b>
<b>4. FORMAT</b>	<b>8</b>
<b>4.1 File Header Record</b>	<b>8</b>
<b>4.2 Field Definitions- Default</b>	<b>9</b>
<b>4.3 Field Definitions - Non-Mode 5 Based</b>	<b>11</b>
<b>4.4 Firmware Binary Data</b>	<b>12</b>
<b>5. APPENDIX</b>	<b>13</b>
<b>5.1 Method of Downloading Firmware</b>	<b>13</b>
<b>5.2 Default Download</b>	<b>13</b>
<b>5.3 Mode and Offset Controlled Download</b>	<b>13</b>
<b>5.3 Determining the Method of Download</b>	<b>15</b>

---

**FIRMWARE DOWNLOAD FORMAT SPECIFICATION**

---

**TABLES**

Table 5-1. Download with Offset, CDB #1 .....	14
Table 5-2. Download with Offset, CDB #2 .....	14
Table 5-3. Download with Offset, CDB #3 .....	15
Table 5-4. Download with Offset, CDB #4 .....	15
Table 5-5. Download with Offset and Save, Last CDB.....	15

---

**FIRMWARE DOWNLOAD FORMAT SPECIFICATION****1. SCOPE**

This document outlines in detail the format specification of download firmware file format used by Unisys to download firmware.

Revision B enables the ability to determine precisely what type of download command sequence a device may require. The modifications for Revision B do not require a new utility for existing download files unless the utility checks for non-blank data after the last Drive ID entry.

This specification is not meant to define vendor unique tape loading or other non-Write Buffer command based loading methods

---

**FIRMWARE DOWNLOAD FORMAT SPECIFICATION**

---

**2. PURPOSE**

The purpose of this document is to standardize the format of the firmware file which will simplify the task of reading the firmware file by the system application programs.

In addition, by standardizing the format of the download firmware file, other future peripheral vendors with download features could be supported by Unisys with minimum modifications required to the system application program. This document supersedes internal document R210.

---

**FIRMWARE DOWNLOAD FORMAT SPECIFICATION****3. PROCEDURE****SCSI Firmware Download File Format**

The firmware data will be provided on a DOS 3.5 " diskette (1.44MB) in binary format. Physically, the media will contain 8K (1K=1024) byte contiguous records. The file shall have two logical record types: a header record which will contain all the information needed to interpret the content and perform some checks on the drives, and a body record which will contain all the firmware code. Physically, the first 8K record on the media will contain the header record. All other 8K records on the media will contain the firmware information. If the firmware size is not a multiple of 8K, then the last 8K record will contain extra storage (00 hex).

**FIRMWARE DOWNLOAD FORMAT SPECIFICATION****4. FORMAT****4.1 File Header Record**

The 8,192 byte firmware file header record format will be as follows:

Field Name	Size (in bytes)	Offset
Customer ID	8	0
Size of header record	8	8
Size of firmware data	8	16
New firmware level	4	24
Vendor ID	8	28
Number of drive ID	4	36
Number of replaceable firmware levels	4	40
Firmware level #1 record (most recent)	8	44
Firmware level #2 record	8	52
... (until the last level)	8	...
Drive ID #1	16	... +8
Drive ID #2	16	... + 16
... (until the last Drive ID #n)	16	... +... + 32
MODE	4	... +... + 48
Mode-Start	1	... +... + 52
Mode-Last	1	... +... + 53
Mode Offset Minimum Increment	8	... +... + 54
Mode Offset Maximum Increment	8	... +... + 62

**Example Header**

0: UNISYS 00008192  
 16: 01048828 2 bFUJI  
 32: TSU 00010001\*  
 48: 2 AM2488D M248  
 64: 8 MODE67000005  
 80: 1201048848

*Note: The \* in the above Header example is supposed to be the value 00 in hex, but there is no way in the ASCII command set to denote that character for the purposes of the example. Also, the above example is for the non-default case of a non-Mode 5 download sequence.*

---

**FIRMWARE DOWNLOAD FORMAT SPECIFICATION****Unused Bytes in the 8K Header**

Typically, there will be a significant amount of unused data at the end of the Header Record. This unused data should be set to 00 hex. Further it is recommended that at least 16 bytes after the last valid entry of header data be set to 00 hex to allow for a break point in the flow of header information.

**4.2 Field Definitions- Default**

The default case for the download sequence uses a single Write Buffer command with Mode 5.

All letter representation is in upper case.

All data is in ASCII format unless otherwise indicated.

**Customer ID**

This field is always set to "UNISYS" and is left justified. Unused portions are ASCII blanks 20h.

EXAMPLE: "UNISYS "

**Size of Header Record**

This field is always set to 8K bytes and is right justified with leading zeros.

EXAMPLE: "00008192"

**Size of Firmware Data**

This field contains the number of decimal bytes (represented in ASCII) of firmware data and is right justified with leading zeros.

EXAMPLE: "00305152"

This value would convert to 4A800 hex bytes for use in the CDB.

**New Firmware Level**

Level of firmware as described in bytes 32 through 35 of SCSI INQUIRY data.

EXAMPLE: "0030"

**Vendor ID**

The name of the drive manufacturer, typically UNISYS for disk products, which corresponds to data returned from bytes 8 through 15 of drive SCSI INQUIRY command. This is specified in the procurement specification for that device and is left justified. Unused portions are ASCII blanks 20h.

EXAMPLE: "UNISYS "

---

**FIRMWARE DOWNLOAD FORMAT SPECIFICATION**

---

**Number of Drive ID**

Number of different drive model numbers that use the same firmware and is right justified with leading zeros.

EXAMPLE: "0002"

**Number of Replaceable Firmware Levels**

Number of all prior firmware levels that were approved for production. This field is right justified with leading zeros.

EXAMPLE: "0002"

**NOTE TO IMPLEMENTORS OF DOWNLOADING UTILITY:**

*If you are downloading a current version on a drive already containing that version of firmware, the use of the "New firmware Level" field data is recommended for verifying the correct version of firmware that is downloaded to the drive.*

**Firmware Level #X Record Format**

8 bytes: numbered 7..0 with left most being byte 7.

byte 7: a format flag meaning,

if = "Hex 00"

"No format needed after the download of this level of firmware", or not a disk device

else

"Format needed after the download" (disk devices only)

byte 6..4:

Reserved (ASCII blanks 20h)

byte 3..0:

firmware level, as it appears in bytes 32 through 35 of the SCSI Inquiry data (Right Justified) for firmware level #X.

*Note: Because the firmware level in this field is used to determine whether or not new firmware needs to be loaded, each new release of drive firmware must be uniquely identifiable in bytes 32 through 35 of the SCSI Inquiry data.*

**Drive ID # Record**

Represents the drive model number or "PRODUCT IDENTIFICATION FIELD" as returned from bytes 16 through 31 of SCSI INQUIRY command and is left justified (ASCII blank 20h for unused portions). Unisys will advise the vendor on required drive ID's (Drive model numbers).

---

**FIRMWARE DOWNLOAD FORMAT SPECIFICATION**

EXAMPLES: "ST41600N" or "ST41600ND"

#### 4.3 Field Definitions - Non-Mode 5 Based

The following field definitions are defined for drives that cannot download using a single ANSI standard Write Buffer command with Mode 5. The use of the fields in this section is optional. These fields should not be used if the product can operate with the default download sequence that uses a single Write Buffer command and Mode 5.

*NOTE TO IMPLEMENTORS OF DOWNLOADING UTILITY:*

*If using the non-Mode 5 method, the download utility will have to get the Mode Information prior to building the CDB.*

**MODE** Record

This is an ASCII 4 byte identifier for the start of the mode definition section of the header record. If present, it indicates the drive does not support the default download method. The remaining MODE supporting records must be used to determine the appropriate Write Buffer command parameters and sequence.

DATA: "MODE"

**Mode-Start**

This ASCII 1 byte value indicates what mode of download should be used for all but the last Write Buffer command, in a multiple Write Buffer series to download microcode.

EXAMPLE: "6"

If the drive cannot support multiple Write Buffer commands firmware download and also cannot support the default of a Mode 5 download, then this field shall contain the same value as the Mode-Last field.

**Mode-Last**

This ASCII 1 byte value indicates what mode of download should be used for the last (or only) Write Buffer command.

EXAMPLE: "7"

**Mode Offset Minimum Increment**

This 8 byte field identifies the minimum incremental number of decimal bytes (represented in ASCII) of firmware data (offset) the drive will accept for multiple Write Buffer commands. It's format is right justified with leading zeroes.

This value is also used to determine the factor by which an offset greater than the minimum is determined. An offset must be an even multiple of the minimum increment. For example, if the minimum increment is 4096, only offsets in multiples of 4096 (e.g. 8192, 12288, or 65536, etc.) may be used.

**FIRMWARE DOWNLOAD FORMAT SPECIFICATION**

EXAMPLE: "00004096"

If the drive does not support the use of offsets in the Write Buffer command, this field shall be set to the Size of Firmware Data field value for consistency.

**Mode Offset Maximum Increment**

This field identifies the maximum incremental number of decimal bytes (represented in ASCII) of firmware data (offset) the drive will accept for multiple Write Buffer commands. It's format is right justified with leading zeroes.

If the drive does not support the use of offsets in the Write Buffer command, this field shall be set to the Size of Firmware Data field value for consistency.

If the drive can accept a single Write Buffer command for download, this field shall contain the Size of Firmware Data field value for consistency.

EXAMPLE: "00305152"

**4.4 Firmware Binary Data**

This record immediately follows the 8K file header. It contains the actual firmware code. The size of the field is set the value in the Size of Firmware Data field.

---

**FIRMWARE DOWNLOAD FORMAT SPECIFICATION****5. APPENDIX**

This section does not apply to the Supplier that is trying to build a header and binary code file. It is informational only and intended to explain how the information in the header is used.

**5.1 Method of Downloading Firmware**

The SCSI Write Buffer command shall be used to download and save the firmware on the drive. There are two methods for downloading the firmware. The preferred capability is for the drive to receive the entire firmware file body as a single Write Buffer command. The second method utilizes offsets and modes as defined in the SCSI-3 Write Buffer command to enable the firmware file to be sent in multiple commands.

**5.2 Default Download**

The following download parameters are defined for backwards compatibility with previous Unisys download utilities, and firmware files. This CDB must be used when the MODE field is not present.

The Write Buffer CDB is set as follows, and the firmware is transferred in a single command:

0x3B	
0x05	05 = Microcode Download
0x00	Buffer ID
0x00	Reserved
0x00	
0x00	
0xXX	Number of bytes sent to drive (3 byte field=Size of Firmware in Hex)
0xXX	
0xXX	
0x00	

**5.3 Mode and Offset Controlled Download**

A download using the Mode and Offset controls requires the utility to generate a CDB based on the MODE and increment values. Consider the following MODE header information:

**FIRMWARE DOWNLOAD FORMAT SPECIFICATION**

Field	Value
MODE	MODE
Mode-Start	6
Mode-Last	7
Minimum Incr.	00004096
Maximum Incr.	00305152

*Note: For the purposes of this example, the maximum increment was set to the Size of Firmware Data value since the drive is capable of a single Write Buffer command download.*

The system for which the download will occur can only send a maximum of 65536 bytes in a SCSI command, despite the drive's ability to receive the firmware in a single command. Since this is the case, the utility will have to generate four CDBs with Mode 6 (the Start value), and another CDB with Mode 7 (Last).

The first 4 CDBs contain the following data:

**Table 5-1. Download with Offset, CDB #1**

0x3B	
0x06	6 = Download with Offset
0x00	Buffer ID
0x00	Buffer Offset
0x00	
0x01	Number of bytes sent to drive (3 byte field)
0x00	
0x00	
0x00	

**Table 5-2. Download with Offset, CDB #2**

0x3B	
0x06	6 = Download with Offset
0x00	Buffer ID
0x01	Buffer Offset
0x00	
0x00	
0x01	Number of bytes sent to drive (3 byte field)
0x00	
0x00	
0x00	

**FIRMWARE DOWNLOAD FORMAT SPECIFICATION****Table 5-3. Download with Offset, CDB #3**

0x3B	
0x06	6 = Download with Offset
0x00	Buffer ID
0x02	Buffer Offset
0x00	
0x00	
0x01	Number of bytes sent to drive (3 byte field)
0x00	
0x00	
0x00	

**Table 5-4. Download with Offset, CDB #4**

0x3B	
0x06	6 = Download with Offset
0x00	Buffer ID
0x03	Buffer Offset
0x00	
0x00	
0x01	Number of bytes sent to drive (3 byte field)
0x00	
0x00	
0x00	

The last CDB contains Mode 7, Download with Offset and Save:

**Table 5-5. Download with Offset and Save, Last CDB**

0x3B	
0x07	7 = Download with Offset and Save
0x00	Buffer ID
0x04	Buffer Offset
0x00	
0x00	
0xA8	Number of bytes sent to drive (3 byte field)
0x00	
0x00	

### 5.3 Determining the Method of Download

This section provides a sample algorithm that the utility implementor can use to determine what type of download should be performed based on the information provided in the Header. It is assumed that before this algorithm is processed that the Header data has been validated with the device, i.e. Unisys name in the header, firmware levels compared, and product model compared.

FIRMWARE DOWNLOAD FORMAT SPECIFICATION

DEFINE VARIABLE "SYSCAP" = The maximum length of a SCSI I/O that the system can send for the Write Buffer Command.

IF THE WORD 'MODE' IS NOT PRESENT IN THE HEADER DATA  
THEN ;DEFAULT DOWNLOAD METHOD MUST BE USED  
    IF 'Size of firmware data' > SYSCAP  
    THEN  
        ERROR "DOWNLOAD IS NOT POSSIBLE"  
    ELSE  
        PERFORM DEFAULT DOWNLOAD (MODE 5, SINGLE COMMAND)

ELSE ;DEFAULT CAN'T BE USED BECAUSE MODE IS PRESENT  
    IF SYSCAP ≥ 'Mode Offset Maximum Increment'  
    THEN  
        IF 'Mode Offset Maximum Increment' = 'Size of firmware data'  
        THEN  
            "DO 1 I/O WITH 'Mode Last' IN THE CDB, AND LENGTH =  
            'Size of firmware data'  
        ELSE  
            "DO 'n' I/Os WITH 'Mode Start' IN THE CDB, AND LENGTH =  
            'Mode Offset Maximum Increment' UNTIL THE LAST I/O WHICH IS  
            EXECUTED WITH 'Mode Last' IN THE CDB. DON'T FORGET THAT  
            THE OFFSETS IN THE CDB HAVE TO INCREMENT WITH EACH I/O."  
    ELSE ; THIS IS THE CASE WHERE DRIVER FIRMWARE IS BIGGER THAN HOST CAP  
        IF 'Mode Offset Minimum Increment' > SYSCAP  
        THEN  
            ERROR "DOWNLOAD IS NOT POSSIBLE. SYSTEM NOT CAPABLE"  
        ELSE  
            "CALCULATE THE LARGEST POSSIBLE BLOCK SIZE THAT IS A  
            MULTIPLE OF THE 'Mode Offset Minimum Increment', BUT IS LESS  
            THAN THE SYSCAP SIZE. THEN-  
            "DO 'n' I/Os WITH 'Mode Start' IN THE CDB, AND LENGTH =  
            'Newly calculated value < SYSCAP' UNTIL THE LAST I/O WHICH IS  
            EXECUTED WITH 'Mode Last' IN THE CDB. DON'T FORGET THAT  
            THE OFFSETS IN THE CDB HAVE TO INCREMENT WITH EACH I/O."

Alanis, Belisario MV

From: Leisz, Frank J TR  
To: Alanis, Belisario MV; Shelton, Phil MV  
Cc: Hale, Wesley MV; Bauman, David A; Stell, Jeffery A  
Subject: Test Unit Ready  
Date: Friday, September 13, 1996 2:09PM

Hi;

In order to accommodate the SSD request for the addition of the Test Unit Ready command, S/W will continue to use the current interface between the DFAST application and the MCP procedure USERMAINTERQUEST.

To invoke the Test Unit Ready command, the User will pass 11 as the function code. The remaining parameter definitions are the same.

For this function code, the MCP will issue the Test Unit Ready Command (00h) and return any detected error condition in the parameter DEVICERD.

Any problems or comments on this interface?

TIA

SEPT 1996  
Information to  
Inventor B. Alanis for  
Integration with MCP  
during DFAST upgrade  
development

(E2)

Alanis, Belisario

MV

---

From: Leisz, Frank J  
To: Alanis, Belisario MV  
Cc: Houseman, Gloria R; Stell, Jeffery A  
Subject: Bootleg 423A MCP (Download Microcode with multi-IO's)  
Date: Tuesday, November 19, 1996 1:56PM

Greetings:

I created a new bootleg 423A MCP, which contains the new download function codes and the Ralph's OST patches.

The name of the bootleg MCP is:

(TRANSFER)SYSTEM/423A/MCP/ASDX/UMR ON TRANSFER located on TRPROGD

Hopefully this helps with available machine time, if there are any problems, please call me at sn-385-4487. TIA

Frank

NOV 1996

Information to  
Inventor B, Alanis for  
Integration with  
Unisys Master Control  
Program (MCP)

(E3)

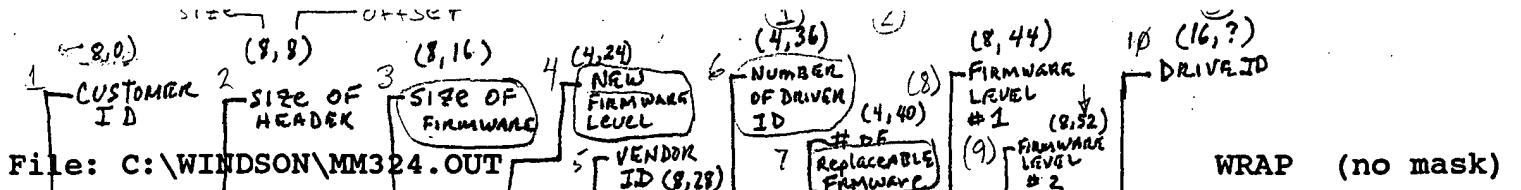
DFAST= ON ENGDATA

File Name	Filekind	Records	Sectors	CreationTime
DFAST	DCALGOLCODE	270	270	08/09/1996
DFAST/NX	DCALGOLCODE	270	270	11/11/1996
DFAST/GAM/DLT/MLTBLK/TUR/NFWCOMPR	DCALGOLCODE	290	297	03/07/1997
DFAST/SPO	DCALGOLCODE	280	288	07/10/1996
DFAST/CHUNK	DCALGOLCODE	270	270	09/23/1996
DFAST/GAMMA/DELTA	DCALGOLCODE	270	270	02/18/1997
DFAST/GAMMA/DELTA/MULTBLK/TUR	DCALGOLCODE	290	297	02/18/1997
DFAST/NXTUR	DCALGOLCODE	290	297	01/28/1997
DFAST/CHUNKY	DCALGOLCODE	286	288	11/27/1996
DFAST/CHUNKY/UNIL0	DCALGOLCODE	286	288	12/02/1996
DFAST/3DIGIT	DCALGOLCODE	269	270	06/27/1996
DFAST/3DIGIT/SYSTPFORDEV	DCALGOLCODE	270	270	06/27/1996
DFAST/CHUNKY2	DCALGOLCODE	290	297	12/12/1996
DFAST/CHUNKY2/UNISYS	DCALGOLCODE	290	297	01/13/1997
DFAST/NXTURFX	DCALGOLCODE	290	297	01/29/1997

Experimental And  
Test Runs by Inventor  
B. Alanis during develop-  
ments For upgrade to  
DFAST,

1996 - 1997

E4



VIEW ASCII Dump Formatted Gather Hex Mask Wordwrap  
COMMANDS F2 F3 F4 F5 F6 goto bookmark F9 search SPACE search again  
↑↓ scroll ALT SHFT menus F10 commands F1 help ESC cancel

MVA154 192.59.221.56

Format Spec. # 3486 8968 Rev B 4/12/96

Symbol /DFAST/1005000

Header work in  
April 1996 by Inventor  
working on upgrade to  
DFAST

E 5

DO YOU WANT TO VIEW THE USER DOCUMENTATION FOR DFAST? 'Y' OR 'N'

Y  
DFAST is a firmware download utility for SCSI disk drives  
and/or SBC Controllers. DFAST requires:

- 1.) Firmware input file on disk or tape; e.g.,

(UCODE)FWFILE/123 ON ANYPACK - if on disk  
or  
(UCODE)FWFILE/123 - if on tape

- 2.) A target (SCSI disk drive or SBC Controller)  
that has been reserved (URed).

- 3.) '<Mix #' AX' when responding to a prompt.

The process follows the following format:

A.) The user is asked to select one of four options:

- 1.) 'Ctlload' - select a SBC unit type.
- 2.) 'Diskload' - select a SCSI disk drive unit type.
- 3.) 'Verifyfile' - check and display firmware file  
header information without  
performing a download to the target.
- 4.) 'Quit' - exit DFAST.

B.) The user is prompted for a firmware file name or  
'Quit' to exit DFAST.

C.) If the file is validated as a firmware file, the new  
firmware level is displayed; otherwise the program  
returns to step 'B'.

D.) If 'Ctlload' or 'Diskload' was entered in step 'A',  
then for every SCSI disk or SBC unit number entered,  
steps 'a' through 'e' below are performed.

a.) Target attributes are obtained and displayed.

b.) If download is not allowed (due to mismatch of file/  
target Firmware level, Product ID or Vendor ID), the  
process stops for that unit, and the user is asked  
to enter another target number of the same unit type  
or 'End' to return to step 'A'.  
otherwise:

c.) If the SCSI drive (not SBC) requires formatting(IVR)  
after the download, the user is consulted to:

- i.) continue with the firmware download -or-
- ii.) stop the process for that unit (to back up data  
on that drive) and select another target of the  
same unit type, or 'End' to return to step 'A'.

d.) After the download is complete and the target has  
sequenced, the INQUIRY command is reissued, and the  
new firmware level of the target is displayed.

e.) The user is prompted to enter another target number  
of the same unit type or 'End' to return to step 'A'.

>>>>>>>>>>>>> ATTENTION <<<<<<<<<<<<

READ the appropriate FCN for this firmware file  
before downloading the firmware

>>>>>>>>>>>>> ATTENTION <<<<<<<<<<<

Do you want to continue? Enter 'YES' or 'NO'

DFAST  
upgrade work  
for Debug &  
Test by  
Inventor Alan S  
in Feb 1997

E6

UNISYS

ENGINEERING INFORMATION RELEASE  
SUMMARY SHEETEIR Number Prt Cmp Rv  
MV - 200523 001 ARel Date: 19990707  
Rev Date:

Product Name: CONTRLR SUPPT  
 Project No : COMMON  
 Type : CHANGE  
 Category : SOFTWARE  
 Eng Affected: MV  
 Mfg Affected:  
 REA(s) :

WORKORDER(s): MV44003

Keywrds:

Reason: UPDATE DFAST LVL 1.004 TO LVL 1.005

Explanation - Full:

RELEASES DFAST MEDIA LEVEL 1.005 TO THE PTDTOQUAL KIT AND  
 RELEASES MEDIA TO POSM. DFAST LEVEL 1.005 ADDS SUPPORT TO THE  
 NX5820 LVD AND SERVER DOWNLOAD.

34925156-003 TO -004

34925149-003 TO -004

EIR Revision: A Est Date: 19990629 Rel Date: 19990707  
 Preparer: C. LAKE EMI Coord:  
 Checker : EMI Info :  
 Init : Mfg Rev :  
 Safety : Other :  
 Authoriz: Other :

Authorizer's Verification Statement:

Lo. Us. Base/Part	List . Fr/To.	Part . Fr/To.	Pic . Fr/To.	Description	. C	BC	. No.	Pg. . SF.	. RM
3492 4639	F	G	*	DFASTMT MED ASSY M42		PL	2	4	
3492 4639-003		F	G	DFASTMT MED ASSY M42		C	I		
3492 4639-004			G	DFASTMT MED ASSY M42	1.0	N	A		
3492 5149	F	G	*	DFASTQIC MED ASSY M17		PL	2	6	
3492 5149-003		F	G	DFASTQIC MED ASSY M17		C	I		
3492 5149-004			G	DFASTQIC MED ASSY M17	1.0	N	A		
3492 5156	F	G	*	DFASTHIC MED ASSY M48		PL	2	8	
3492 5156-003		F	G	DFASTHIC MED ASSY M48	LV	1.0	C	I	
3492 5156-004			G	DFASTHIC MED ASSY M48	LV	1.0	N	A	
3493 0222	E	G	*	DFASTHIC MSTR MED M48		TL	2	10	
3493 0222-003		E	G	DFASTHIC MSTR MED M48		C	I		
3493 0222-004			G	DFASTHIC MSTR MED M48		N	A		
3493 0230	E	G	*	DFASTHIC LAB TXT M48		TL	2	12	
3493 0230-003		E	G	DFASTHIC LAB TXT M48		C	I		
3493 0230-004			G	DFASTHIC LAB TXT M48		N	A		
3493 1014	E	F	*	DFASTQIC MSTR MED M17		TL	2	14	
3493 1014-003		E	F	DFASTQIC MSTR MED M17		C	I		
3493 1014-004			F	DFASTQIC MSTR MED M17		N	A		
3493 1022	E	F	*	DFASTQIC LAB TXT M17		TL	2	16	
3493 1022-003		E	F	DFASTQIC LAB TXT M17		C	I		
3493 1022-004			F	DFASTQIC LAB TXT M17		N	A		

EXHIBIT

E-8

UNISYS

ENGINEERING INFORMATION RELEASE  
SUMMARY SHEET

EIR Number Prt Cmp Rv

MV - 200523 001 A

Rel Date: 19990707

Rev Date:

Product Name: CONTRLR SUPPT  
RCC: 4 Priority: A

Page 2 of 23

Lo. Us. Base/Part	List . Fr/To.	Part . Fr/To.	Pic . Fr/To.	Description	C . C	BC St	No. Pg. SF.	RM
3493 1030	E	F	*	DFASTMT MSTR MED M42	TL	2	18	
3493 1030-003		E	F	DFASTMT MSTR MED M42	C	I		
3493 1030-004			F	DFASTMT MSTR MED M42	N	A		
3493 1048	E	F	*	DFASTMT LAB TXT M42	TL	2	20	
3493 1048-003		E	F	DFASTMT LAB TXT M42	C	I		
3493 1048-004			F	DFASTMT LAB TXT M42	N	A		

U N I S Y S

ENGINEERING INFORMATION RELEASE  
SUMMARY SHEET  
CONTINUATION

EIR Number Prt Cmp Rv  
MV - 200523 001 A  
-----  
Rel Date: 19990707  
Rev Date:

Product Name: CONTRLR SUPPT  
RCC: 4 Priority: A

Page 3 of 23

EIR CONTINUATION REV: A  
RELEASES DFAST MEDIA LEVEL 1.005 TO THE PTDTOQUAL KIT AND  
RELEASES MEDIA TO POSM. DFAST LEVEL 1.005 ADDS SUPPORT TO THE  
NX5820 LVD AND SERVER DOWNLOAD.

34925156-003 TO -004

34925149-003 TO -004

34924639-003 TO -004

SCRAP OLD PART NUMBERS.

U N I S Y S

ENGINEERING INFORMATION RELEASE  
DETAIL SHEETEIR Number MV - 200523 001  
Prt Cmp A  
Page 4 of 23

Base Number	List Fr/To	Pic Fr/To	Description	No Sh			
				BM	Pr	RCC	PL
3492 4639	F	G	* * DFASTM MED ASSY M42				4 1

## CEDB CHANGES:

## BASE NUMBER RELATED CHANGES:

	From	To
CHG Range:	000-003	000-004
CHG EPC:	*	COMMON
ADD PFC:		SOFTWR

## PART NUMBER RELATED CHANGES:

Part Fr/To	Description	CC	St
		---	---
3492 4639-003 F G	DFASTM MED ASSY M42	C	I

	From	To
CHG Desc:	DFASTM LV:1.004 ..	DFASTM MED ASSY M42
CHG Status:	A	I
ADD Newpn:		3492 4639-004

## DIFFERENCES BETWEEN 3492 4639-003 REPLACED-INACTIVE BY 3492 4639-004

Find	Ch	Alt	From	To
				---
1				
CHG			Part: 3937 6975-000 MAGNETIC TAPE, 1200 FT. BLANK MEDIA ASSY	3937 7007-000 MAG TAPE 3600 FOOT

## DEL R001

Ref: 3493 1030-003  
DFASTM MSTR MED M42  
DFASTM MSTR MED M42

## R002

CHG Ref: 3493 1048-003  
DFASTM LAB TXT M42  
DFASTM LAB TXT M42

3493 1030-004  
DFASTM MSTR MED M42  
DFASTM MSTR MED M42

U N I S Y S

ENGINEERING INFORMATION RELEASE  
DETAIL SHEETEIR Number Prt Cmp Rv  
MV - 200523 001 A  
Page 5 of 23

Base Number	List Fr/To	Pic Fr/To	Description					No Sh
				BM	Pr	RCC	-----	
3492 4639	F G	* *	DFASTMT MED ASSY M42	PL	A	4	-----	2

## DIFFERENCE REPORT Continued

Find	Ch	Alt	From	To
ADD R003	-----	-----	-----	-----
	Ref:			3493 1048-004
				DFASTMT LAB TXT M42
				DFASTMT LAB TXT M42

## PART NUMBER RELATED CHANGES:

Part Fr/To	Description			CC	St
		-----	-----		
3492 4639-004	G	DFASTMT MED ASSY M42	1.005	N	A

From	To
ADD Desc:	-----
ADD Status:	DFASTMT MED ASSY M42
	A

## MANUFACTURING STATUS INFORMATION

## DISPOSITION OF PARTS:

<input type="checkbox"/> SCRAP OLD PARTS	<input type="checkbox"/> DEPLET OLD STOCK	<input type="checkbox"/> REWORK
<input type="checkbox"/> INCORP ON OPEN ORDERS	<input type="checkbox"/> DRAW MATL FOR EVALUATN	<input type="checkbox"/> OTHER
<input type="checkbox"/> USE UNTIL NEW PARTS AVAIL	<input type="checkbox"/> INCORP NEXT TIME BLD	

## IMPLEMENT ON ASSEMBLIES/PARTS:

<input type="checkbox"/> SERIAL NUMBER:	<input type="checkbox"/> IN MFG	<input type="checkbox"/> IN TEST	<input type="checkbox"/> AT VENDOR	<input type="checkbox"/> IN FIELD
<input type="checkbox"/> REWORK:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

## COMMENTS:

U N I S Y S

ENGINEERING INFORMATION RELEASE  
DETAIL SHEETEIR Number MV - 200523 001  
Prt Cmp A  
Page 6 of 23

Base Number	List Fr/To	Pic Fr/To	Description	No			
				BM	Pr	RCC	Sh
3492 5149	F	G	* * DFASTQIC MED ASSY M17	PL	A	4	1

CEDB CHANGES:

BASE NUMBER RELATED CHANGES:

	From	To
CHG Range:	000-003	000-004
CHG EPC:	*	COMMON
ADD PFC:		SOFTWR

PART NUMBER RELATED CHANGES:

Part Fr/To	Description	CC	St
		---	---
3492 5149-003	F G DFASTQIC MED ASSY M17	C	I

	From	To
CHG Desc:	DFASTQIC LV: 1.004 ..	DFASTQIC MED ASSY M17
CHG Status:	A	I
ADD Newpn:		3492 5149-004

DIFFERENCES BETWEEN 3492 5149-003 REPLACED-INACTIVE BY 3492 5149-004

Find	Ch	Alt	From	To
DEL	R001			---
			Ref: 3493 1014-003	
			DFASTQIC MSTR MED M17	
			DFASTQIC MSTR MED M17	
	R002			
CHG			Ref: 3493 1022-003	3493 1014-004
			DFASTQIC LAB TXT M17	DFASTQIC MSTR MED M17
			DFASTQIC LAB TXT M17	DFASTQIC MSTR MED M17
ADD	R003			
			Ref:	3493 1022-004
				DFASTQIC LAB TXT M17
				DFASTQIC LAB TXT M17

U N I S Y S

ENGINEERING INFORMATION RELEASE  
DETAIL SHEETEIR Number Prt Cmp Rv  
MV - 200523 001 A  
Page 7 of 23

Base Number	Fr/To	Fr/To	Description	BM	Pr	RCC	No Sh
3492 5149	F	G	* * DFASTQIC MED ASSY M17	PL	A	4	2

## PART NUMBER RELATED CHANGES:

Part	Fr/To	Description	CC	St
3492 5149-004	G	DFASTQIC MED ASSY M17	1.005	N A

From To  
--- ---  
ADD Desc: DFASTQIC MED ASSY M17 1.005  
ADD Status: A

## MANUFACTURING STATUS INFORMATION

## DISPOSITION OF PARTS:

<input type="checkbox"/> SCRAP OLD PARTS	<input type="checkbox"/> DEPLET OLD STOCK	<input type="checkbox"/> REWORK
<input type="checkbox"/> INCORP ON OPEN ORDERS	<input type="checkbox"/> DRAW MATL FOR EVALUATN	<input type="checkbox"/> OTHER
<input type="checkbox"/> USE UNTIL NEW PARTS AVAIL	<input type="checkbox"/> INCORP NEXT TIME BLD	

## IMPLEMENT ON ASSEMBLIES/PARTS:

<input type="checkbox"/> SERIAL NUMBER:	<input type="checkbox"/> IN MFG	<input type="checkbox"/> IN TEST	<input type="checkbox"/> AT VENDOR	<input type="checkbox"/> IN FIELD
<input type="checkbox"/> REWORK:				

COMMENTS:

U N I S Y S

ENGINEERING INFORMATION RELEASE  
DETAIL SHEETEIR Number MV - 200523 001  
Prt Cmp Page 8 of 23 Rv A

Base Number	List Fr/To	Pic Fr/To	Description	No			
				BM	Pr	RCC	Sh
3492 5156	F G	* *	DFASTHIC MED ASSY M48	PL	A	4	1

CEDB CHANGES:

BASE NUMBER RELATED CHANGES:

	From	To
CHG Range:	000-003	000-004
CHG EPC:	*	COMMON
ADD PFC:		SOFTWR

PART NUMBER RELATED CHANGES:

Part Fr/To	Description	CC	St
		-----	-----
3492 5156-003	F G DFASTHIC MED ASSY M48 LV 1.00	C	I

	From	To
CHG Desc:	DFASTHIC LV:1.004 ..	DFASTHIC MED ASSY M48 LV 1.00
CHG Status:	A	I
ADD Newpn:		3492 5156-004

DIFFERENCES BETWEEN 3492 5156-003 REPLACED-INACTIVE BY 3492 5156-004

Find	Ch	Alt	From	To
DEL	R001			---
			Ref: 3493 0222-003	
			DFASTHIC MSTR MED M48	
			DFASTHIC MSTR MED M48	
	R002			
CHG			Ref: 3493 0230-003	3493 0222-004
			DFASTHIC LAB TXT M48	DFASTHIC MSTR MED M48
			DFASTHIC LAB TXT M48	DFASTHIC MSTR MED M48
	ADD R003			
			Ref:	3493 0230-004
				DFASTHIC LAB TXT M48
				DFASTHIC LAB TXT M48

U N I S Y S

ENGINEERING INFORMATION RELEASE  
DETAIL SHEETEIR Number Prt Cmp Rv  
MV - 200523 001 A  
Page 9 of 23

Base Number	List Fr/To	Pic Fr/To	Description	No				
				BM	Pr	RCC	Sh	
3492 5156	F	G	* *	DFASTHIC MED ASSY M48	PL	A	4	2

## PART NUMBER RELATED CHANGES:

Part Fr/To	Description	CC		St	
		---	---	N	A
3492 5156-004	G	DFASTHIC MED ASSY M48 LV 1.005	---	N	A

From	To
ADD Desc:	DFASTHIC MED ASSY M48 LV 1.005
ADD Status:	A

## MANUFACTURING STATUS INFORMATION

## DISPOSITION OF PARTS:

<input type="checkbox"/> SCRAP OLD PARTS	<input type="checkbox"/> DEPLET OLD STOCK	<input type="checkbox"/> REWORK
<input type="checkbox"/> INCORP ON OPEN ORDERS	<input type="checkbox"/> DRAW MATL FOR EVALUATN	<input type="checkbox"/>
<input type="checkbox"/> USE UNTIL NEW PARTS AVAIL	<input type="checkbox"/> INCORP NEXT TIME BLD	<input type="checkbox"/> OTHER

## IMPLEMENT ON ASSEMBLIES/PARTS:

<input type="checkbox"/> SERIAL NUMBER:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> REWORK:	<input type="checkbox"/> IN MFG	<input type="checkbox"/> IN TEST	<input type="checkbox"/> AT VENDOR	<input type="checkbox"/> IN FIELD	<input type="checkbox"/>

COMMENTS:

U N I S Y S

ENGINEERING INFORMATION RELEASE  
DETAIL SHEETEIR Number MV - 200523 001  
Prt Cmp A  
Page 10 of 23

Base Number	Fr/To	Fr/To	Description	BM	Pr	RCC	No Sh
3493 0222	E G	*	DFASTHIC MSTR MED M48	TL	A	4	1

## CEDB CHANGES:

## BASE NUMBER RELATED CHANGES:

	From	To
CHG Range:	000-003	000-004
CHG EPC:	*	COMMON
ADD PFC:		SOFTWR

## PART NUMBER RELATED CHANGES:

Part Fr/To	Description	CC	St
3493 0222-003 E G	DFASTHIC MSTR MED M48	C	I

	From	To
ADD Desc:		DFASTHIC MSTR MED M48
CHG Status:	A	I
ADD Newpn:		3493 0222-004

Part Fr/To	Description	CC	St
3493 0222-004 G	DFASTHIC MSTR MED M48	N	A

	From	To
ADD Desc:		DFASTHIC MSTR MED M48
ADD Status:		A

U N I S Y S

ENGINEERING INFORMATION RELEASE  
DETAIL SHEETEIR Number Prt Cmp Rv  
MV - 200523 001 A  
Page 11 of 23

Base Number	List Fr/To	Pic Fr/To	Description	BM	Pr	RCC	No Sh
3493 0222	E	G	* * DFASTHIC MSTR MED M48	--	TL	A	4 2

## MANUFACTURING STATUS INFORMATION

## DISPOSITION OF PARTS:

<input type="checkbox"/> SCRAP OLD PARTS	<input type="checkbox"/> DEPLET OLD STOCK	<input type="checkbox"/> REWORK
<input type="checkbox"/> INCORP ON OPEN ORDERS	<input type="checkbox"/> DRAW MATL FOR EVALUATN	<input type="checkbox"/>
<input type="checkbox"/> USE UNTIL NEW PARTS AVAIL	<input type="checkbox"/> INCORP NEXT TIME BLD	<input type="checkbox"/> OTHER

## IMPLEMENT ON ASSEMBLIES/PARTS:

<input type="checkbox"/> SERIAL NUMBER:	<input type="checkbox"/>	<input type="checkbox"/> IN MFG	<input type="checkbox"/> IN TEST	<input type="checkbox"/> AT VENDOR	<input type="checkbox"/> IN FIELD
<input type="checkbox"/> REWORK:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

COMMENTS:

U N I S Y S

ENGINEERING INFORMATION RELEASE  
DETAIL SHEETEIR Number Prt Cmp Rv  
MV - 200523 001 A  
Page 12 of 23

Base Number	List Fr/To	Pic Fr/To	Description					No Sh
				BM	Pr	RCC	TL	
3493 0230	E G	* *	DFASTHIC LAB TXT M48				4	1

CEDB CHANGES:

BASE NUMBER RELATED CHANGES:

	From	To
CHG Range:	000-003	000-004
CHG EPC:	*	COMMON
ADD PFC:		LABEL

PART NUMBER RELATED CHANGES:

Part Fr/To	Description	CC	St
		-----	-----
3493 0230-003	E G	DFASTHIC LAB TXT M48	C I

	From	To
ADD Desc:	-----	DFASTHIC LAB TXT M48
CHG Status:	A	I
ADD Newpn:		3493 0230-004

Part Fr/To	Description	CC	St
		-----	-----
3493 0230-004	G	DFASTHIC LAB TXT M48	N A

	From	To
ADD Desc:	-----	DFASTHIC LAB TXT M48
ADD Status:		A

UNISYS

ENGINEERING INFORMATION RELEASE  
DETAIL SHEETEIR Number Prt Cmp Rv  
MV - 200523 001 A  
Page 13 of 23

Base Number	Fr/To	Fr/To	List Pic	Description	BM	Pr	RCC	No Sh
3493 0230	E	G	*	DFASTHIC LAB TXT M48	TL	A	4	2

## MANUFACTURING STATUS INFORMATION

## DISPOSITION OF PARTS:

<input type="checkbox"/> SCRAP OLD PARTS	<input type="checkbox"/> DEPLET OLD STOCK	<input type="checkbox"/> REWORK
<input type="checkbox"/> INCORP ON OPEN ORDERS	<input type="checkbox"/> DRAW MATL FOR EVALUATN	<input type="checkbox"/>
<input type="checkbox"/> USE UNTIL NEW PARTS AVAIL	<input type="checkbox"/> INCORP NEXT TIME BLD	<input type="checkbox"/> OTHER

## IMPLEMENT ON ASSEMBLIES/PARTS:

<input type="checkbox"/> SERIAL NUMBER:	<input type="checkbox"/> IN MFG	<input type="checkbox"/> IN TEST	<input type="checkbox"/> AT VENDOR	<input type="checkbox"/> IN FIELD
<input type="checkbox"/> REWORK:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

COMMENTS:

U N I S Y S

ENGINEERING INFORMATION RELEASE  
DETAIL SHEETEIR Number MV - 200523 001  
Prt Cmp A  
Page 14 of 23

Base Number	List Fr/To	Pic Fr/To	Description					No Sh
				BM	Pr	RCC	---	
3493 1014	E F	* *	DFASTQIC MSTR MED M17	---	TL	A	4	1

## CEDB CHANGES:

## BASE NUMBER RELATED CHANGES:

	From	To
CHG Range:	000-003	000-004
ADD Med Cd:		CT
CHG EPC:	*	COMMON
ADD PFC:		SOFTWR

## PART NUMBER RELATED CHANGES:

Part Fr/To	Description		
		CC	St
3493 1014-003	E F	DFASTQIC MSTR MED M17	C I

	From	To
ADD Desc:		DFASTQIC MSTR MED M17
CHG Status:	A	I
ADD Newpn:		3493 1014-004

Part Fr/To	Description		
		CC	St
3493 1014-004	F	DFASTQIC MSTR MED M17	N A

	From	To
ADD Desc:		DFASTQIC MSTR MED M17
ADD Status:		A

U N I S Y S

ENGINEERING INFORMATION RELEASE  
DETAIL SHEETEIR Number Prt Cmp Rv  
MV - 200523 001 A  
Page 15 of 23

Base Number	List Fr/To	Pic Fr/To	Description					No Sh
				BM	Pr	RCC	TL	
3493 1014	E F	* *	DFASTQIC MSTR MED M17				4	2

## MANUFACTURING STATUS INFORMATION

## DISPOSITION OF PARTS:

<input type="checkbox"/> SCRAP OLD PARTS	<input type="checkbox"/> DEPLET OLD STOCK	<input type="checkbox"/> REWORK
<input type="checkbox"/> INCORP ON OPEN ORDERS	<input type="checkbox"/> DRAW MATL FOR EVALUATN	<input type="checkbox"/>
<input type="checkbox"/> USE UNTIL NEW PARTS AVAIL	<input type="checkbox"/> INCORP NEXT TIME BLD	<input type="checkbox"/> OTHER

## IMPLEMENT ON ASSEMBLIES/PARTS:

<input type="checkbox"/> SERIAL NUMBER:	<input type="checkbox"/>	<input type="checkbox"/> IN MFG	<input type="checkbox"/> IN TEST	<input type="checkbox"/> AT VENDOR	<input type="checkbox"/> IN FIELD
<input type="checkbox"/> REWORK:	<input type="checkbox"/>				

## COMMENTS:

UNISYS

ENGINEERING INFORMATION RELEASE  
DETAIL SHEETEIR Number Prt Cmp Rv  
MV - 200523 001 A  
Page 16 of 23

Base Number	List Fr/To	Pic Fr/To	Description	No			
				BM	Pr	RCC	Sh
3493 1022	E F	* *	DFASTQIC LAB TXT M17	TL	A	4	1

CEDB CHANGES:

BASE NUMBER RELATED CHANGES:

	From	To
CHG Range:	000-003	000-004
ADD Med Cd:		CT
CHG EPC:	*	COMMON
ADD PFC:		LABEL

PART NUMBER RELATED CHANGES:

	Part Fr/To	Description	CC	St
3493 1022-003	E F	DFASTQIC LAB TXT M17	C	I

	From	To
ADD Desc:		DFASTQIC LAB TXT M17
CHG Status:	A	I
ADD Newpn:		3493 1022-004

	Part Fr/To	Description	CC	St
3493 1022-004	F	DFASTQIC LAB TXT M17	N	A

	From	To
ADD Desc:		DFASTQIC LAB TXT M17
ADD Status:		A

U N I S Y S

ENGINEERING INFORMATION RELEASE  
DETAIL SHEETEIR Number MV - 200523 001  
Prt Cmp A  
Page 17 of 23

Base Number	Fr/To	Fr/To	Pic	Description	BM	Pr	RCC	No Sh
3493 1022	E	F	*	DFASTQIC LAB TXT M17	TL	A	4	2

## MANUFACTURING STATUS INFORMATION

## DISPOSITION OF PARTS:

<input type="checkbox"/> SCRAP OLD PARTS	<input type="checkbox"/> DEPLET OLD STOCK	<input type="checkbox"/> REWORK
<input type="checkbox"/> INCORP ON OPEN ORDERS	<input type="checkbox"/> DRAW MATL FOR EVALUATN	<input type="checkbox"/>
<input type="checkbox"/> USE UNTIL NEW PARTS AVAIL	<input type="checkbox"/> INCORP NEXT TIME BLD	<input type="checkbox"/> OTHER

## IMPLEMENT ON ASSEMBLIES/PARTS:

<input type="checkbox"/> SERIAL NUMBER:	<input type="checkbox"/> IN MFG	<input type="checkbox"/> IN TEST	<input type="checkbox"/> AT VENDOR	<input type="checkbox"/> IN FIELD
<input type="checkbox"/> REWORK:				

COMMENTS:

U N I S Y S

ENGINEERING INFORMATION RELEASE  
DETAIL SHEETEIR Number Prt Cmp Rv  
MV - 200523 001 A  
Page 18 of 23

Base Number	List Fr/To	Pic Fr/To	Description	No			
				BM	Pr	RCC	Sh
3493 1030	E F	* *	DFASTMT MSTR MED M42	TL	A	4	1

CEDB CHANGES:

BASE NUMBER RELATED CHANGES:

	From	To
CHG Range:	000-003	000-004
CHG EPC:	*	COMMON
ADD PFC:		SOFTWR

PART NUMBER RELATED CHANGES:

Part Fr/To	Description	CC		St	
		--	--	C	I
3493 1030-003	E F	DFASTMT MSTR MED M42			

	From	To
ADD Desc:		DFASTMT MSTR MED M42
CHG Status:	A	I
ADD Newpn:		3493 1030-004

Part Fr/To	Description	CC		St	
		--	--	N	A
3493 1030-004	F	DFASTMT MSTR MED M42			

	From	To
ADD Desc:		DFASTMT MSTR MED M42
ADD Status:		A

U N I S Y S

ENGINEERING INFORMATION RELEASE  
DETAIL SHEETEIR Number Prt Cmp Rv  
MV - 200523 001 A  
Page 19 of 23

Base Number	Fr/To	Fr/To	List Pic	Description	BM	Pr	RCC	No Sh
3493 1030	E	F	*	DFASTMT MSTR MED M42	TL	A	4	2

## MANUFACTURING STATUS INFORMATION

## DISPOSITION OF PARTS:

<input type="checkbox"/> SCRAP OLD PARTS	<input type="checkbox"/> DEPLET OLD STOCK	<input type="checkbox"/> REWORK
<input type="checkbox"/> INCORP ON OPEN ORDERS	<input type="checkbox"/> DRAW MATL FOR EVALUATN	<input type="checkbox"/>
<input type="checkbox"/> USE UNTIL NEW PARTS AVAIL	<input type="checkbox"/> INCORP NEXT TIME BLD	<input type="checkbox"/> OTHER

## IMPLEMENT ON ASSEMBLIES/PARTS:

SERIAL NUMBER:	<input type="checkbox"/>	IN MFG	<input type="checkbox"/> IN TEST	<input type="checkbox"/> AT VENDOR	<input type="checkbox"/> IN FIELD
REWORK:	<input type="checkbox"/>				

COMMENTS:

U N I S Y S

ENGINEERING INFORMATION RELEASE  
DETAIL SHEETEIR Number Prt Cmp Rv  
MV - 200523 001 A  
Page 20 of 23

Base Number	List Fr/To	Pic Fr/To	Description	No			
				BM	Pr	RCC	Sh
3493 1048	E F	* *	DFASTMT LAB TXT M42	TL	A	4	1

CEDB CHANGES:

BASE NUMBER RELATED CHANGES:

	From	To
CHG Range:	000-003	000-004
ADD Med Cd:		MT
CHG EPC:	*	COMMON
ADD PFC:		LABEL

PART NUMBER RELATED CHANGES:

Part Fr/To	Description	CC		St	
		—	—	—	—
3493 1048-003	E F	DFASTMT LAB TXT M42	C	I	

	From	To
ADD Desc:	—	DFASTMT LAB TXT M42
CHG Status:	A	I
ADD Newpn:		3493 1048-004

Part Fr/To	Description	CC		St	
		—	—	—	—
3493 1048-004	F	DFASTMT LAB TXT M42	N	A	

	From	To
ADD Desc:	—	DFASTMT LAB TXT M42
ADD Status:		A

U N I S Y S

ENGINEERING INFORMATION RELEASE  
DETAIL SHEETEIR Number Prt Cmp Rv  
MV - 200523 001 A  
Page 21 of 23

Base Number	List Fr/To	Pic Fr/To	Description	No			
				BM	Pr	RCC	Sh
3493 1048	E F	* *	DFASTMT LAB TXT M42	--	--	--	--

## MANUFACTURING STATUS INFORMATION

## DISPOSITION OF PARTS:

<input type="checkbox"/> SCRAP OLD PARTS	<input type="checkbox"/> DEPLET OLD STOCK	<input type="checkbox"/> REWORK
<input type="checkbox"/> INCORP ON OPEN ORDERS	<input type="checkbox"/> DRAW MATL FOR EVALUATN	<input type="checkbox"/>
<input type="checkbox"/> USE UNTIL NEW PARTS AVAIL	<input type="checkbox"/> INCORP NEXT TIME BLD	<input type="checkbox"/> OTHER

## IMPLEMENT ON ASSEMBLIES/PARTS:

<input type="checkbox"/> SERIAL NUMBER:	<input type="checkbox"/> IN MFG	<input type="checkbox"/> IN TEST	<input type="checkbox"/> AT VENDOR	<input type="checkbox"/> IN FIELD
<input type="checkbox"/> REWORK:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

COMMENTS:

U N I S Y S

ENGINEERING INFORMATION RELEASE  
REPORTS/ADDENDUM

EIR Number Prt Cmp Rv

MV - 200523 001 A

-----

Rel Date: 19990707

Rev Date:

---

Product Name: CONTRLR SUPPT  
RCC: 4 Priority: A

---

Page 22 of 23

TOP UNITS AFFECTED:

---

Top	Unit	Part	Sta	Bom	DCL	Description
2346	1809-999	A	SU	MV		KIT: PTDTOQUAL
3445	3068-000	I	SU	MV		KIT, SCSI FIRMWARE DOWNLOAD
3445	3068-001	I	SU	MV		KIT, SCSI FIRMWARE DOWNLOAD
3445	3068-002	I	SU	MV		KIT, SCSI FIRMWARE DOWNLOAD
3445	3068-003	A	SU	MV		KIT, SCSI FIRMWARE DOWNLOAD
4490	2542-000	A	SU	MV		KIT, FCN-USR4000-010 A-MT
4490	2567-000	A	SU	MV		INDEX, FCN USR4000-010

---

U N I S Y S

ENGINEERING INFORMATION RELEASE  
APPENDIX: STRUCTURE CHANGESEIR Number Prt Cmp Rv  
MV - 200523 001 A  
Page 23 of 23

3492 4639-003 DFASTMT MED ASSY M42

DFASTMT MED ASSY M42

DIFFERENCES BETWEEN 3492 4639-003 REPLACED-INACTIVE BY 3492 4639-004

	Find	Ch	Alt	From	To
CHG		1		3937 6975-000	3937 7007-000
DEL Ref	R001			3493 1030-003	
CHG Ref	R002			3493 1048-003	3493 1030-004
ADD Ref	R003				3493 1048-004

3492 5149-003 DFASTQIC MED ASSY M17

DFASTQIC MED ASSY M17

DIFFERENCES BETWEEN 3492 5149-003 REPLACED-INACTIVE BY 3492 5149-004

	Find	Ch	Alt	From	To
DEL Ref	R001			3493 1014-003	
CHG Ref	R002			3493 1022-003	3493 1014-004
ADD Ref	R003				3493 1022-004

3492 5156-003 DFASTHIC MED ASSY M48

DFASTHIC MED ASSY M48 LV 1.00

DIFFERENCES BETWEEN 3492 5156-003 REPLACED-INACTIVE BY 3492 5156-004

	Find	Ch	Alt	From	To
DEL Ref	R001			3493 0222-003	
CHG Ref	R002			3493 0230-003	3493 0222-004
ADD Ref	R003				3493 0230-004

# EXHIBIT

E-9

```

%% VERSION 90.032.001
%% SET STACK LIMIT 100
%% RESET LIST XREF
%% SET VERSION 01.005.000
%% SET OMIT
% PATCHFOR DOWNLOAD IN MULTIPLE CHUNKS
%%%%%%%%%%%%%%00014000
%
% Class | Unisys
% | 00015000
% | 00016000
% | 00017000
% | This material is restricted and proprietary to the 00018000
% | Unisys Corporation and is not to be reproduced, shown, 00019000
% ## ## | or disclosed outside the Unisys Corporation. Customer 00020000
% ## ## | Services Engineering restricted and proprietary data 00021000
% ## ## | is furnished solely for use by Unisys personnel in 00022000
% ## ## | servicing customer's equipment. 00023000
% ## | 00024000
% ## | 00025000
% ## ## | This document is the property of and shall be returned 00026000
% ## ## | to Unisys Corporation, One Unisys Place, Detroit, MI 00027000
% ## ## | 48232. 00028000
% | 00029000
% | 00030000
% Material | Copyright (C) 1994.
% | 00031000
% | 00032000
% | 00033000
% | 00034000
% | 00035000
%%%%%%%%%%%%%%00036000
$> PAGE
%%%%%%%%%%%%%%00038000
%
% TARGET / FW LOAD
% | 00039000
% | 00040000
% | 00041000
% | 00042000
% | 00043000
%%%%%%%%%%%%%%00044000
4/94 Initial version. Will download A-code files to SBC controllers 00045000
and SCSI Disk Drives. 00046000
4/94 Changes - Qualification phase for official CSPO release 01.001. 00047000010010001
5/94 Add 'Express Mode' for engineering (DISKS only) release 01.002. 00047200010020003
6/94 Change location of some display statements. --- release 01.002. 00047500010020002
10/94 Fixed seg array error in Verifyfile procedure. release 01.002. 00047600010020004
00048000
%%%%%%%%%%%%%%00049000
% DOWNLOAD FIRMWARE to A-SERIES TARGETS (DFAST)
% -----
% Utility's Part Number: 3492 4639
% | 00052000
% | 00053000
% | 00054000
% | 00055000
% | 00056000
% | 00057000
% | 00058000
% | 00059000
% | 00060000
% | 00061000
% | 00062000
% | 00063000
% | 00064000
% | 00065000
% | 00066000
% | 00067000
% | 00068000
% | 00069000
% | 00070000
% | 00071000
% | 00072000
% | 00073000
% | 00074000
% | 00075000
% | 00076000
% | 00077000
%
Load:
-----
To operate, two basic elements must be present: an SBC controller or 00070000
SCSI disk drive, and a microcode file on disk or tape. The microcode 00071000
must reside on a unit served by a different controller or disk than the 00072000
one being initialized. If a "critical unit" exists on the string served 00073000
by the controller being initialized and there is only one path to that 00074000
critical unit, the system will reject the attempt to download the 00075000
microcode. 00076000
00077000

```

D1

1) Step one is to determine the code file's capability by having the user enter a file name. Normal Family Substitution rules are in effect. If the file cannot be found, the user is prompted to enter another file name. The file name may include an "ON <family>" in the file declaration if it resides on disk (e.g., (UCODE)XYZ/123 ON MYPACK). If the file resides on tape, the file name only would be entered (e.g., SCZFRM).

If the code file is not a valid SBC or disk drive Acode file, the file is rejected and the user is prompted to enter another file name.

2) Step two is to determine the SBC controller or drive capability. 00089000  
The user enters a SBC or drive number and the utility reads the unit's 00090000  
attributes. If the SBC or drive number does not represent the correct 00091000  
target, or if the target attributes do not match those of the Acode 00092000  
file, the user is prompted to enter a new target number or 'Quit'. The 00093000  
SBC controller or drive must be reserved. 00094000

>>> There may be no way to prove that the SBC controller controls any given drive or that it is the only path to the drive or that another path exists! The User should do an "OL" on the ODT to verify the paths available for the target. 00096000 00097000 00098000 00099000

Verifyfile: 00102000010020003  
----- 00103000010020003  
00104000

Verify is simpler than Load since no SBC or drive is involved. The 00105000 user answers the file name prompt and the Verify routine used by Load is 00106000 called to generate a report of the attributes associated with that file. 00107000

```

Verify allows the user to cycle through multiple, potential A-code 00109000
files until 'Quit' is entered. 00110000
$$ POP OMIT 00111000
BEGIN 00112000
***** Structure Generating Declarations 00113000
* 00114000
***** 00115000
DEFINE 00116000
OVHD = 12 # * At front of each code segment 00117000
,XSTATBYTES = 254 # * Read Unit Status return length 00118000
,EIGHTK = 8192 # 00119000
,ENDSGD = # 00120000
,SYSCAP = 393216 # 00120500010040006
,MAXELEMENTS = 1 # * MAX ELEMENT PER DIMENSION 0012060001.005.000
,MAXROWS = 48 # * MAX NUMBER OF ROWS 0012080001.005.000
;

```

FILE		00123000	
CODE(	KIND	= TAPE,                   % UNKNOWN TYPE	00124000
\$8888	LABELKIND	= UNLABELLED,           % READ ONLY	00125000
FILEUSE		= IN,	00126000
OPTIONAL		= TRUE,	00127000
NEWFILE		= FALSE,	00128000
DEPENDENTSPECS		= TRUE)	00129000
,LINE(	KIND	= PRINTER,	00130000
FILEUSE		= OUT,	00131000
FRAMESIZE		= 8,	00132000
MAXRECSIZE		= 132)	00133000
,RMT(	KIND	= REMOTE,	00134000
FILEUSE		= IO,	00135000
BLOCKSTRUCTURE		= EXTERNAL,	00136000
FRAMESIZE		= 8,	00137000
MAXRECSIZE		= 132)	00138000

```

DIRECT EBBCDIC ARRAY
  IMLBUF2      [0:0,0:0]          % To be resized
, IMLBUF3      [0:0,0:0]          % To be resized
, IMLBUF      [0:0]               .
;                                         00140000
                                         00141000
                                         001420001.0040006
                                         0014220001.005.001
                                         00142500010040006
                                         00143000

```

## 2 DIMENSIONAL BUFFER

D 2

EBCDIC ARRAY		00144000
,CODEREC [0:0]	\$ CODE record	00145000
,ETSTMP [0:23]	\$ Timestamp data	00146000
,FCUSTID [0:7]	\$ FW file customer ID	00147000
,FNEWFWLEVEL [0:3]	\$ FW file new firmware level	00148000
,FVENDORID [0:7]	\$ FW file vendor ID	00149000
,OLDFWLEVEL [0:3]	\$ FW level before download	00150000
,HDR [0:0]	\$ CODE record zero	00151000
,HDPRESULT [0:29]		00152000
,INBUF [0:131]	\$ User input buffer	00153000
,L [0:131]	\$ Formatting space	00154000
,SCR2 [0:131]	\$ Scratch 2	00154500010040006
,SCR [0:131]	\$ Scratch space	00155000
,TTL [0:89]	\$ CODE.TITLE	00156000
,SHORTBUF [0:XSTATBYTES]	\$ UMR Interface calls	00157000
,ESHORTBUF [0:XSTATBYTES]	\$ ASCII to EBCDIC conversion	00158000
,SAVEFWLVL [0:7]	\$ FW file data area of FW level	00159000
,OEM [0:71]	\$ DRIVE TYPE	00159500010020004
;		00160000
REAL ARRAY		00161000
,RCODE[0]	= CODEREC	00162000
,RHDR[0]	= HDR	00163000
,TSTMP[0]	= ETSTMP	00164000
,X[0:0]		00165000
;		00165500010020004
HEX ARRAY		00166000
,HSCR[0]	= SCR;	00167000
PICTURE		00168000
,DATENTIME(N/99I99I99" @ "N:99I99);		00169000
TRUTHSET		00170000
,NUMBERS ("0123456789")		00171000
,HEXDIGITS (NUMBERS OR "ABCDEF")		00172000
,ENDER (" .");		00173000
TRANSLATETABLE		0017320001.005.000
,UPCASER(EBCDIC TO EBCDIC,		0017340001.005.000
"abcdefghijklmnopqrstuvwxyz" TO		00174000
"ABCDEFGHIJKLMNOPQRSTUVWXYZ");		00175000
;		00176000
,Real and Integer Variables		00177000
;		00178000
REAL		00179000
,FKAIW	\$ These Reals are either bit buckets or other values which	00180000
,MRD	\$ would not survive integerization.	00181000
,SYSTEMTYPE	\$ FILEKIND Attribute Info Word	00181200010020004
,VAL	\$ Maintenance Result Descriptor bits	00181400010020004
;	\$ All, A14, A16, A18, A19 are valid	00182000010020004
;	\$ Value of hex to decimal conversion	00183000
;		00184000
INTEGER		00185000
,CODEEOF	\$ FILEKIND Attribute Info Word	00186000
,CODEMRS	\$ Maintenance Result Descriptor bits	00187000
,CTLUNIT	\$ All, A14, A16, A18, A19 are valid	00188000
,FHDRBYTES	\$ Value of hex to decimal conversion	00189000
,FCODEBYTES	\$ Code size (# bytes) in firmware file	00190000
;		00191000
;		00192000
;		00193000
;		00194000
;		00195000
;		00196000
;		00197000010010001
;		00197500010020004
;		00198000
;		00199000
;		00200000
;		00201000
;		00202000
;		00203000
;		00204000
;		00205000

```

,FUNCTION          % Operating mode for this run          00206000
,LL               % Number of bytes remaining in PL      00207000
,SEGINX           % Segment starting record index      00208000
,SEGMR5           % Segment maxrecsize                  00209000
,SOFAR             % Bytes moved to IMLEBUF so far          00210000
,TIMER             % Drop dead timer (intervals)           00211000
,NUMREPFWLVL5    % Number of replaceable firmware levels 00212000
,DRIVEIDLOC      % Array index to FW drive IDs           00213000
,NUMDRIVEID      % Number of drive IDs valid for the code 00214000
,OPTODO            % Opcode parameter to UMR                00215000
,NUMBROFIOS      % # of IO's to do for multi chunk dwld 00215200010040006
,SIZEOFLSTIO      % Size of last IO to do multi chunk dnlld 00215400010040006
,NROWS             % Number of rows required for download 00215500010040006
,MROWS             % Number of MOD rows as residual rows 0021560001.005.000
;
;                  00216000010040006
;                  00217000
;*****00218000
%               Boolean Variables                  00219000
;*****00220000
BOOLEAN           00221000
    CTLASSIGNED      % CTL assigned                  00222000
    ,SDMASSIGNED      % SCSI disk drive assigned      00223000
    ,DEBUG             % Internal debug mode          00224000
    ,ODTMODE           % User is at a console or terminal 00225000
    ,ATODT             % User is at console           00226000
    ,SKIPODTPRINT     % Don't print ODT display       00226500010010001
    ,EXPRESSMODE      % Omit many displays (for engineering) 00226600010020003
    ,VERIFYONLY        % Don't build FW image in buffer 00226800010020004
    ,FIRSTTIME         % 0022690001.005.000
;
;                  00227000
;                  00228000
;*****00229000
%               Structure Independent Defines 00230000
;*****00231000
DEFINE             00232000
    ADD               = REPLACE PL:PL BY #          00233000
    ,B                 = BOOLEAN #                  00234000
    ,BLANK            = REPLACE L BY " " FOR 22 WORDS # 00235000
    ,C                 = , #                         00236000
    ,CLEARINBUF       = REPLACE INBUF BY " " FOR 12 WORDS # 00237000
    ,NUL              = 48"00" #                  00238000
    ,P                 = POINTER #                  00239000
    ,PUT(N)           = REPLACE PL:L[N] BY #        00240000
    ,SHOW(S,TF)        = BEGIN
                        START S;
                        SHOWIT(TF);
                        END #
    ,START             = REPLACE PL:L BY #        00241000
;
;                  00242000
;                  00243000
;                  00244000
;                  00245000
;                  00246000
;*****00247000
%               Formatting constants
    ,TXTLOC30          = 30 #                      00248000
    ,TXTLOC22          = 22 #                      00249000
    ,TXTLOC15          = 15 #                      00250000
    ,ERRLOC            = (TXTLOC30+2) #          00251000
;
;                  00252000
;                  00253000
;*****00254000
%               Function values
    ,LOADCTLV          = 1 #                      00255000
    ,LOADENVV          = 2 #                      00256000
    ,VERIFYV            = 3 #                      0025700001.005.000
;
;                  00258000
;                  00259000
;*****00260000
%               Code File Header (most recent version)
%   Elements for all fields include their location and their length.
;
;*****00261000
    ,CUSTIDLOC          = 0 #                      00262000
    ,CUSTIDLNG          = 8 #                      00263000
    ,CUSTIDEXP          = "FUJITSU " #          00264000
    ,CUSTIDEXP          = "UNISYS " #          00264500010040006
    ,HDRECSIZELOC        = 8 #                      00265000010040006
    ,HDRECSIZELNG        = 8 #                      00266000
;
;                  00267000

```

```

,FWCODESIZELOC      = 16 #          00268000
,FWCODESIZELNG     = 8 #          00269000
,NEWFWLEVELLOC     = 24 #          00270000
,NEWFWLEVELNG      = 4 #          00271000
,VENDORIDLOC       = 28 #          00272000
,VENDORIDLNG       = 8 #          00273000
,NUMDRIVEIDLOC    = 36 #          00274000
,NUMDRIVEIDLNG    = 4 #          00275000
,NUMREPFWLVLLOC   = 40 #          00276000
,NUMREPFWLVLNG    = 4 #          00277000
,OLDFWLEVELNG     = 8 #          00278000
,DRIVEIDLNG       = 16 #          00279000
% ,VENDORIDEXP     = "FUJITSU " #  00279500010040006
% ,VENDORIDEXP     = "UNISYS " #   00280000010040006
% *,NOTESTRING      = "SBCfirmware." #
% ,TVENDORID        = ESHORTBUF [10] # % FOR 8 00281000
% ,TPRODID          = ESHORTBUF [18] # % FOR 16 00282000
% ,TPRODREVLVL     = ESHORTBUF [34] # % FOR 4 00283000
% ,TOEMID           = ESHORTBUF [54] # % FOR 8 00284000
% ,TSERVOREVLVL    = ESHORTBUF [14] # % FOR 8 00284500010020004
% ,TSERVOREVLVL    = ESHORTBUF [14] # % FOR 8 0028460001.005.000
;

% User Maint Requests

DEFINE
  ASSIGNUNITV        = 0 #          00285000
  UNASSIGNUNITV      = 1 #          00286000
  LOADSLAVEIMLV     = 2 #          00287000
  READERERRORLOGV   = 3 #          00288000
  ATTRIBUTESV        = 4 #          00289000
  ASSIGNNCTLV        = 5 #          00290000
  UNASSIGNNCTLV      = 6 #          00291000
  MODESENSE          = 7 #          00292000
  DOWNLOADMODE4      = 8 #          00293000
  DOWNLOADMODE6      = 9 #          00294000
  DOWNLOADMODE7      = 10 #         00295000
  TESTUNITREADY      = 11 #         00296000
  INQUIRY             = 12 #         00297000
;

LIBRARY
  MCP(  FUNCTIONNAME  = "MCPSUPPORT .",
        LIBACCESS      = BYFUNCTION); 00298000
  00299000
  00300000
  00301000
  00302000
  00303000
  00304000
  00305000
  00306000
  00307000
  00308000
  00309000
  00310000
  00311000
  00312000
  00313000
  00314000
  00315000
  00316000
  00317000
  00318000
  00319000
  00320000
  00321000
  00322000
  00323000
  00324000
  00325000
  00326000
  00327000
  00328000
  00329000

INTEGER PROCEDURE USERMINTREQUEST(
  UNITNUMBER, FUNCTIONCODE, IOLENGTH, BUFFEROFFSET, BUFFERID, 00330000
  BUFFER, MRD, DEVICERD); 003304000
VALUE UNITNUMBER, FUNCTIONCODE, IOLENGTH, BUFFEROFFSET, BUFFERID; 003305000
INTEGER UNITNUMBER, FUNCTIONCODE, IOLENGTH, BUFFEROFFSET, BUFFERID, MRD; 003306000
EBCDIC ARRAY BUFFER[*], DEVICERD[*]; 003307000
*****003308000
*. Description: 003309000
*   This procedure is a Library-exported interface that supports 00331000
*   SCSI and IPI-3 I/O utilities (e.g., microcode download utility). 003311000
*. Parameters: 003312000
*   UNITNUMBER        External unit number of the device being 003313000
*   accessed. The MCP will determine whether a 003314000
*   port, CTL, or a device is being accessed using 003315000
*   data from the I/O unit tables. 003316000
*   FUNCTIONCODE:    Case index to be executed in the interface. 003317000
*   0: Assign Unit 003318000
*   1: Unassign Unit 003319000
*   2: Do FW Load (Load Slave IML) 003320000
*   3: Read Extended Status (Read Error Log) 003321000
*   4: Obtain Disk ID (Report Attributes) 003322000
*   5: Assign CTL 003323000
*   6: Unassign CTL 003324000
*   IOLENGTH          Bytes 003325000
*   BUFFEROFFSET      Used by SCSI interface to allow non- 003326000
*   transmittable data to precede actual data 003327000
*   in the buffer. 003328000
*   003329000

```

DS

```

%*                                Not required for IPI.          00330000
%  BUFFERID      Not used by SCSI.                      00331000
%*                                Not required for IPI.          00332000
%  BUFFER       Direct EBCDIC array to be used as the source 00333000
%                                of the data and the repository for extended 00334000
%                                parameter handling.          00335000
%  MRD          Maintenance Result Descriptor bits. Some bits 00336000
%                                refer to HDPRESULT as returned to PTD. 00337000
%  DEVICERD     Status and Sense data                   00338000
% Assumptions:          00339000
% This routine protects the integrity of the system from a poorly 00340000
% written application-level calling program.                      00341000
% Results:          00342000
%   0:             Function succeeded.                  00343000
%   >0:            Function failed. See SHOWRSLT for details. 00344000
% Locks:          00345000
%   None.          00346000
%%%%%%%%%%%%%% 00347000
LIBRARY MCP;          00348000
                      00349000
LABEL
  MAINLOOP          00350000
  ,GRANDIXT;        00351000
$ PAGE BEGINSEGMENT 00352000
%%%%%%%%%%%%%% 00353000
% CONVERT HEX TO DECIMAL          00353040010020004
%%%%%%%%%%%%%% 00353060010020004
PROCEDURE CONVERTHEXTODEC; 00353080010020004
  BEGIN          00353100010020004
    00353120010020004
    INTEGER I;          00353140010020004
    POINTER PTR;        00353160010020004
    00353162010020004
    REAL
    TEMP
    ,VAL;
    VAL := 0;          00353164010020004
    PTR := SCR;        00353166010020004
    00353168010020004
    00353180010020004
    00353200010020004
    00353220010020004
    00353280010020004
    00353300010020004
    FOR I := 1 STEP -1 UNTIL 0 DO          00353320010020004
      BEGIN          00353340010020004
        IF PTR IN NUMBERS THEN          00353360010020004
          TEMP := INTEGER(PTR,1) * 16**I
        ELSE
          CASE REAL(PTR,1) OF
            BEGIN
              "A" : TEMP := 10 * 16**I;
              "B" : TEMP := 11 * 16**I;
              "C" : TEMP := 12 * 16**I;
              "D" : TEMP := 13 * 16**I;
              "E" : TEMP := 14 * 16**I;
              "F" : TEMP := 15 * 16**I;
            ELSE : ;
          END;
        PTR      := * + 1;
        VAL      := * + TEMP;
      END;
    REPLACE OEM BY " " FOR 12 WORDS;          00353440010020004
    REPLACE OEM BY VAL FOR * DIGITS;          00353460010020004
    00353480010020004
    00353500010020004
    00353520010020004
    00353540010020004
    00353560010020004
    00353580010020004
    00353600010020004
    00353620010020004
    00353640010020004
    00353660010020004
    00353680010020004
    00353685010020004
    00353690010020004
    00353695010020004
    00353700010020004
    % PROCEDURE CONVERTHEXTODEC          00354000
    % Common I/O Routines Block          00355000
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 00356000
PROCEDURE SHOWIT(DSP);          00357000
  VALUE          DSP;
  BOOLEAN        DSP;
----- 00358000
----- 00359000
----- 00360000

```

```

% Purpose: 00361000
% Show L on LINE. 00362000
%
% Parameters: 00363000
% DSP True: Caller wants a copy sent to user. 00364000
% 00365000
% 00366000
%
% Globals: 00367000
% ODTMODE True: ODT is normal location of user. 00368000
% False: Terminal is. 00369000
%
% Assumptions: 00370000
% L has message text. 00371000
% PL points to next place to add text. 00372000
% L will be blanked out upon exit. 00373000
% 00374000
%-----00375000
BEGIN 00376000
  DEFINE SEGMENT = #; 00377000
  00378000
  IF NOT SKIPODTPRINT THEN 00378500010010001
    WRITE(LINE,OFFSET(PL),L); % Log the transaction 00379000010010001
  IF DSP THEN % Send a copy to the user 00381000
    IF ODTMODE THEN % at the console 00382000
      BEGIN 00383000
        REPLACE PL BY NUL; 00384000
        DISPLAY(L); 00385000
        WAIT((3)); % Slow down ODT traffic 00386000
      END 00387000
    ELSE % at his terminal 00388000
      WRITE(RMT,OFFSET(PL),L); 00389000
    BLANK; 00390000
  END; % Show It 00391000
  00392000
  00393000
  00394000
  PROCEDURE PROMPT; 00395000
%-----00396000
% Purpose: 00397000
% Display text to the user and expect a response. 00398000
%
% Globals: 00399000
% ODTMODE How we are to perform this action. 00400000
%
% Assumptions: 00401000
% L contains the output message. 00402000
% PL points to the next valid character in L. 00403000
% L needs to contain the input response. 00404000
% Both the output and input will be logged. 00405000
% The output will be upcased after logging in its original form. 00406000
% The output will be prescanned and deblanked using PL and LL. 00407000
% Under NO condition should PROMPT call SHOWIT. SHOWIT blanks out 00408000
% L as it exits, thus destroying all of our user input. 00409000
%-----00410000
BEGIN 00411000
  BOOLEAN BRSLT; 00412000
  00413000
  IF ODTMODE THEN 00414000
    BEGIN 00415000
      WRITE(LINE,OFFSET(PL),L); % Log it 00416000
      REPLACE PL BY NUL; 00417000
      ACCEPT(L); % Both output and input space 00418000
      SCAN L FOR LL:132 UNTIL = NUL; 00419000
      LL := 132 - LL; 00420000
      WRITE(LINE,LL,L); % Log it again 00421000
    END 00422000
  ELSE % Terminal mode 00423000
    BEGIN 00424000
      WRITE(LINE,OFFSET(PL),L); % Log it 00425000
      WRITE(RMT,OFFSET(PL),L); % Tell user 00426000
      BRSLT := READ(RMT,80,L); 00427000
      IF BRSLT THEN % Error, maybe even ?END 00428000
        00429000
        00430000

```

```

GO GRANDXIT; 00431000
PL := L[LL:=REAL(BRSLT).[47:20]]; 00432000
WRITE(LINE,LL,L);           % Log it again 00433000
END; 00434000
00435000
REPLACE L BY L FOR LL WITH UPCASER; 00436000
SCAN PL:L FOR LL:LL WHILE = " "; 00437000
END; % Prompt 00438000
$$ ENDSEGMENT 00439000
$$ PAGE BEGINSEGMENT 00440000
#####00441000
% Debugging Module 00442000
#####00443000
% Contents: 00444000
% SHOWMRDBITS 00445000
% SHOWHDPRESULT 00446000
% SHOWRLST 00447000
% SHOWATTRIBUTES 00448000
#####00449000
PROCEDURE SHOWMRDBITS; 00450000
-----
% Purpose: 00451000
% Expand the Maintenance Result Descriptor bits following a call to 00452000
% USERMAINTREQUEST. 00453000
% 00454000
% 00455000
% Globals: 00456000
% MRD      The Maintenance Result Descriptor 00457000
% 00458000
% Assumptions: 00459000
% The MRD is a packed bit field. 00460000
% We don't know how to decode this yet. 00461000
% This entire procedure will be replaced with defines, case statements, 00462000
% value arrays, etc. 00463000
% For the moment, just dump the MRD in hex. 00464000
#####00465000
BEGIN 00466000
  DEFINE SEGMENT = #; 00467000
 00468000
  REPLACE SCR BY MRD FOR 6; 00469000
  PUT(TXTLOC30) 00470000
    "MRD = ", 00471000
    HSCR FOR 12 WITH HEXTOEBCDIC; 00472000
    SHOWIT(TRUE); 00473000010040006
 00474000
END; % Show MRD bits 00475000
00476000
PROCEDURE SHOWHDPRESULT; 00477000
-----
% Purpose: 00478000
% Expand the HDPRESULT coming out of the USERMAINTREQUEST. 00479000
% 00480000
% Globals: 00481000
% HDPRESULT      A byte array containing RMM data. 00482000
% 00483000
% Assumptions: 00484000
% We think that only 12 bytes of data are valid. We may have 00485000
% gotten an interface error if our array is too short. 00486000
% We don't know the details of what is in this thing yet. 00487000
% For now, we will just format the first 12 bytes in hex. 00488000
% 00489000
%-----00490000
BEGIN 00491000
 00492000
  REPLACE SCR BY HDPRESULT FOR 12; 00493000
  PUT(TXTLOC30) 00494000
    "HDP = ", 00495000
    HSCR[00] FOR 12 WITH HEXTOEBCDIC, 00496000
    " ", 00497000
    HSCR[12] FOR 16 WITH HEXTOEBCDIC; 0049800001.005.000
    SHOWIT(TRUE); 00499000010040006
 00500000
END; % Show HDP result 00501000

```

```

PROCEDURE SHOWRSLT(RSLT,FCN);
  VALUE          RSLT,FCN;
  INTEGER        RSLT,FCN;
-----
% Purpose:
%   Expand the integer value contained in RSLT from a USERMAINTREQUEST
%   into understandable text.
%
% Parameters:
%   RSLT      The procedure value returned from USERMAINTREQUEST.
%   FCN > 0   What function were we trying to do?
%   FCN < 0   Elapsed time in Read Attributes.
%
% Assumptions:
%   If RSLT = 0, the result was clean.
%   If RSLT > 0, it is a CASE code for the actual error.
%   We can use this to make global decisions in addition to
%   just printing out the value.
%
BEGIN
  INTEGER MM,SS;
  LABEL XIT;
  BLANK;                                % Often follows a PROMPT
  IF FCN < 0 THEN                         % Format elapsed time
BEGIN
  FCN := ABS(FCN);
  MM := FCN DIV 60;
  SS := FCN MOD 60;
  START
    MM FOR 2 DIGITS, ":" ,
    SS FOR 2 DIGITS, " Read Attributes: " ,
    SS FOR 2 DIGITS, " Target: " ,
    CTLUNIT FOR * DIGITS,
    " storing firmware.";
  GO XIT;
END
ELSE
CASE FCN OF
BEGIN
ASSIGNUNITV:
  START "Assign Unit: " ,   CTLUNIT FOR * DIGITS;
UNASSIGNUNITV:
  START "Unassign Unit: " , CTLUNIT FOR * DIGITS;
LOADSLAVEIMLV:
  START "Load Slave FW: " , CTLUNIT FOR * DIGITS;
READERRORLOGV:
  START "Read Error Log: " , CTLUNIT FOR * DIGITS;
ATTRIBUTEV:
  START "Read Attributes: " , CTLUNIT FOR * DIGITS;
ASSIGNCTLV:
  START "Assign CTL: " ,   CTLUNIT FOR * DIGITS;
UNASSIGNCTLV:
  START "Unassign CTL: " , CTLUNIT FOR * DIGITS;
END;
PUT(TXTLOC22)
  "RSLT = " ,
  RSLT FOR * DIGITS,
  ":" ;
CASE RSLT OF
BEGIN
0: ADD "No error";
1: ADD "Unit not supported (Vendor ID not UNISYS)";
3: ADD "Unit is not reserved (URed)";
5: ADD "Illegal/undefined function code specified";
7: ADD "I/O error. See MRD & DEVICERD parameters";
9: ADD "I/O length exceeds buffer's capacity";
11: ADD "UMR can't assign unit (reason unknown)";
13: ADD "Target is marked 'broken' by MCP";
00502000
00503000
00504000
00505000
00506000
00507000
00508000
00509000
00510000
00511000
00512000
00513000
00514000
00515000
00516000
00517000
00518000
00519000
00520000
00521000
00522000
00523000
00524000
00525000
00526000
00527000
00528000
00529000
00530000
00531000
00532000
00533000
00534000
00535000
00536000
00537000
00538000
00539000
00540000
00541000
00542000
00543000
00544000
00545000
00546000
00547000
00548000
00549000
00550000
00551000
00552000
00553000
00554000
00555000
00556000
00557000
00558000
00559000
00560000
00561000
00562000
00563000
00564000
00565000
00566000
00567000
00568000
00569000
00570000
00571000
00572000

```

```

15: ADD "Target is marked 'in use' by MCP"; 00573000
17: ADD "Invalid target number was specified"; 00574000
19: ADD "Unit not owned (FREE) by this partition"; 00575000
21: ADD "I/O cannot be initiated"; 00576000
23: ADD "UMR Can't determine target 'unit type'"; 00577000
25: ADD "UMR issued I/O to unassigned target"; 00578000
27: ADD "Unit not present (FREEd)"; 00579000
29: ADD "Can't find usable path to target"; 00580000
31: ADD "Illegal Unassign"; 00580200010040006
33: ADD "IO length exceeds Max IO of 393216"; 00580400010040006
ELSE: ;
END;

XIT:
SHOWIT(TRUE);

END; % Show result 00587000
00588000
00589000
00590000
00591000
00592000
00593000
00594000
00595000
00596000
00597000
00598000
00599000
00600000
00600100010020004
00600200010020004
00600250010020004
00600300010020004
00600400010020004
00600500010020004
00600600010020004
00600700010020004
00601000
00602000
00602100010020003
00602150010020003
00602200010020003
00602250010020003
00602300010020003
00602350010020004
00602400010020004
00602450010020003
00602500010020003
00602550010020003
00602600010020003
00602650010020003
00603000010020003
00604000010020003
00605000010020003
00606000010020004
00607000010020004
00608000010020003
00609000010020003
00609500010020003
00610000
00611000
00612000
0061202001.005.000
0061204001.005.000
0061206001.005.000
0061208001.005.000
0061210001.005.000
0061212001.005.000
0061214001.005.000
0061216001.005.000

-----  

% Purpose:
% Expand the text in the SHORTBUF following a call to USERMAINTREQUEST
% with the ATTRIBUTESV parameter.
%-----  

% Globals:
% SHORTBUF Contains the text results of ATTRIBUTES command.

BEGIN

REPLACE SCR [0] BY " " FOR 2; 00600100010020004
REPLACE POINTER(X[0]) BY SHORTBUF [54] FOR 1; 00600200010020004
REPLACE SCR[0] BY POINTER(X[0],4) FOR 2 WITH HEXTODEBCDIC; 00600250010020004
% REPLACE SCR BY "C8";
% SHOW (" SCR = " C SCR FOR 2, TRUE);

CONVERTHEXTODEC;

REPLACE ESHORTBUF [0] BY SHORTBUF [0] FOR SIZE (ESHORTBUF)
      WITH ASCIITOEBDIC;
IF EXPRESSMODE THEN      % Don't display - only print out
  BEGIN
    SHOW ("TARGET Attributes: "      C
          "VENDOR ID - "      C
          TVENDORID FOR VENDORIDLNG C ".", FALSE);
    SHOW ("PRODUCT ID - "      C TPRODID FOR DRIVEIDLNG
          "- "      C OEM FOR 3 C ".", FALSE);
    SHOW ("FW LEVEL - "      C
          TPRODREV_LVL FOR NEWFWLEVELNG C ".", FALSE);
  END
  ELSE
    % Display and print
    BEGIN
      SHOW ("TARGET Attributes: "      C
            "VENDOR ID - "      C
            TVENDORID FOR VENDORIDLNG C ".", TRUE);
      SHOW ("PRODUCT ID - "      C TPRODID FOR DRIVEIDLNG
            "- "      C OEM FOR 3 C ".", TRUE);
      SHOW ("FW LEVEL - "      C
            TPRODREV_LVL FOR NEWFWLEVELNG C ".", TRUE);
    END;
END; % Show attributes

PROCEDURE SHOWINQUIRYDATA;
%-----  

% Purpose:
% Expand the text in the SHORTBUF following a call to USERMAINTREQUEST
% with the INQUIRY parameter.
%-----  

% Globals:
% SHORTBUF Contains the text results of ATTRIBUTES command.

```

三八

```

BEGIN 00626000
LABEL LOOPER; 00627000
00628000
00629000
00630000
%*IF NOT ATODT THEN    * AT A TERMINAL 00631000
%*BEGIN
SHOW ("DFAST is a firmware download utility for SCSI disk drives",TRUE);00631000
SHOW ("and/or SBC Controllers.  DFAST requires:      ",TRUE);00632000
SHOW ("      ",TRUE);00633000
SHOW ("      1.) Firmware input file on disk or tape; e.g.,      ",TRUE);00634000
SHOW ("      ",TRUE);00635000
SHOW ("      (UCODE)FWFILE/123 ON ANYPACK - if on disk      ",TRUE);00636000
SHOW ("      or      ",TRUE);00637000
SHOW ("      (UCODE)FWFILE/123      - if on tape      ",TRUE);00638000
SHOW ("      ",TRUE);00639000
SHOW ("      2.) A target (SCSI disk drive or SBC Controller)      ",TRUE);00640000
SHOW ("      that has been reserved (URed).      ",TRUE);00641000
SHOW ("      ",TRUE);00642000
SHOW ("      3.) '<Mix #> AX' when responding to a prompt.      ",TRUE);00643000
SHOW ("      ",TRUE);00644000
IF NOT ATODT THEN 00645000
    WAIT ((5));
SHOW ("The process follows the following format:      ",TRUE);00646000
SHOW ("      ",TRUE);00647000
SHOW ("A.) The user is asked to select one of four options:      ",TRUE);00648000
SHOW ("      1.) 'Ctlload'      - select a SBC unit type.      ",TRUE);00650000010020003
SHOW ("      2.) 'Devload'      - select a SCSI disk drive unit type.      ",TRUE);0065100001.005.000
SHOW ("      3.) 'Verifyfile'      - check and display firmware file      ",TRUE);00652000010020003
SHOW ("      header information without      ",TRUE);00653000010020003
SHOW ("      performing a download to the target.",TRUE);00654000010020003
SHOW ("      4.) 'Quit'      - exit DFAST.      ",TRUE);00655000010020003
SHOW ("B.) The user is prompted for a firmware file name or      ",TRUE);00656000
SHOW ("      'Quit' to exit DFAST.      ",TRUE);00657000
SHOW ("C.) If the file is validated as a firmware file, the new      ",TRUE);00658000
SHOW ("      firmware level is displayed; otherwise the program      ",TRUE);00659000
SHOW ("      returns to step 'B'.      ",TRUE);00660000
IF NOT ATODT THEN 00661000
    WAIT ((5));
SHOW ("D.) If 'Ctlload' or 'Diskload' was entered in step 'A',      ",TRUE);00663000
SHOW ("      then for every SCSI disk or SBC unit number entered,      ",TRUE);00664000
SHOW ("      steps 'a' through 'e' below are performed.      ",TRUE);00665000
SHOW ("a.) Target attributes are obtained and displayed.      ",TRUE);00666000
SHOW ("b.) If download is not allowed (due to mismatch of file/      ",TRUE);00667000
SHOW ("      target Firmware level, Product ID or Vendor ID), the      ",TRUE);00668000
SHOW ("      process stops for that unit, and the user is asked      ",TRUE);00669000
SHOW ("      to enter another target number of the same unit type      ",TRUE);00670000
SHOW ("      or 'End' to return to step 'A'.      ",TRUE);00671000
SHOW ("      otherwise:      ",TRUE);00672000
SHOW ("c.) If the SCSI drive (not SBC) requires formatting(IVR)      ",TRUE);00673000010020004
SHOW ("      after the download, the user is consulted to:      ",TRUE);00674000
SHOW ("      i.) continue with the firmware download -or-      ",TRUE);00675000
SHOW ("      ii.) stop the process for that unit (to back up data      ",TRUE);00676000
SHOW ("      on that drive) and select another target of the      ",TRUE);00677000
SHOW ("      same unit type, or 'End' to return to step 'A'.      ",TRUE);00678000
SHOW ("d.) After the download is complete and the target has      ",TRUE);00679000
SHOW ("      sequenced, the INQUIRY command is reissued, and the      ",TRUE);00680000
SHOW ("      new firmware level of the target is displayed.      ",TRUE);00681000
SHOW ("e.) The user is prompted to enter another target number      ",TRUE);00682000
SHOW ("      of the same unit type or 'End' to return to step 'A'.",TRUE);00683000
IF NOT ATODT THEN 00684000
    WAIT ((5));
%*END 00685000
$$ SET OMIT 00686000
    ELSE 00687000
        % AT AN A-SERIES CONSOLE 00688000
    BEGIN 00689000
        SHOW ("      QUIT the program.      ", TRUE); 00690000
        SHOW ("      another device type or Firmware file or      ", TRUE); 00691000
        SHOW ("      or SBC) or exit to outer block to select      ", TRUE); 00692000
        SHOW ("      f.) Loop back for next unit of same type (disk      ", TRUE); 00693000
        SHOW ("      e.) If error free, the process completes.      ", TRUE); 00694000
        SHOW ("      the process for that unit.      ", TRUE); 00695000
        SHOW ("      either continue with downloading or stop      ", TRUE); 00696000

```

```

SHOW ("      after download, the user is consulted to  ", TRUE); 00697000
SHOW ("  d.) If the drive (not SBC) requires formatting ", TRUE); 00698000
SHOW ("      (IVR) otherwise                                ", TRUE); 00699000010020004
SHOW ("      Vendor ID) the process stops for that unit.", TRUE); 00700000
SHOW ("      of unit/file Firmware level, Product ID or ", TRUE); 00701000
SHOW ("  c.) If download is not allowed (due to mismatch", TRUE); 00702000
SHOW ("  b.) Unit information is obtained (INQUIRY).      ", TRUE); 00703000
SHOW ("  a.) The unit must be RESERVED (URed).           ", TRUE); 00704000
SHOW ("  for every Disk or SBC unit entered by the user: ", TRUE); 00705000
SHOW ("  If the firmware file is read in properly, then  ", TRUE); 00706000
SHOW ("The process follows the following format:          ", TRUE); 00707000
SHOW ("      ", TRUE); 00708000
SHOW ("  2.) SCSI Disk Drive or SBC Controller          ", TRUE); 00709000
SHOW ("      e.g., (UCODE)FWFILE/123 ON ANYPACK          ", TRUE); 00710000
SHOW ("  1.) Firmware input file                      ", TRUE); 00711000
SHOW ("  drives and/or SBC Controllers. DFAST requires: ", TRUE); 00712000
SHOW ("DFAST is a firmware download utility for SCSI disk", TRUE); 00713000
END;
$$ POP OMIT
00714000
00715000
00730000
00730200010020002
00730400010020002
00731000
00732000
00733000
00734000
00735000
00736000
00737000
00738000
00739000
00740000
00740200010020003
00740400010020003
00741000
00742000
00743000
00744000
00744200010020003
00744400010020003
00745000
00746000
00747000
00748000010020003
00749000010020004
00750000010020004
00751000010020004
00752000010020004
00753000010020004
00754000010020003
00755000
00756000010020003
00756500010010001
00757000010020004
00758000010020004
00759000010020004
00760000010020004
00761000010020004
00761100010010001
00761200010020004
00761300010020004
00761400010020004
00761500010020004
00761600010020004
00762000010020003
00763000
00764000
00765000
00766000
00766500010020003
00767000
END;  % IF NOT EXPRESSMODE

```

```

SHOW ("Unit " C CTLUNIT FOR * DIGITS C " will need formatting " C      00770000010010001
      "(IVR) after firmware is loaded!", TRUE);                      00771000010020004
SHOW ("", TRUE);                                                 00778000010010001
SHOW ("DO YOU STILL WANT TO CONTINUE WITH THE DOWNLOAD? ", TRUE); 00779000010010001
START "Answer 'YES' or 'NO'";                                         00780000010010001
PROMPT;                                                               00781000
IF PL = "Y" THEN                                                     00782000
  OKTOFORMAT := TRUE                                              00783000
ELSE                                                               00784000
  OKTOFORMAT := FALSE;                                            00785000
00786000
END;  & PROCEDURE OKTOFORMAT                                         00787000
00788000
00789000
00790000
00791000
00792000
00793000
00794000
00795000
00796000
00797000
00798000
00799000
00800000
00801000
00802000
00803000
00804000
00805000
00806000
00807000
00808000
00809000
00810000
00810200010020003
00810400010020003
00810600010020003
00811000010020003
00813000
00814000
00815000
00815200010020003
00815400010020003
00815600010020003
00816000010020003
00817000
00818000
00819000
00820000
00821000
00822000
00823000
00824000
00825000
00826000
00827000
00828000
00829000
00830000
00831000
00832000
00833000
00834000
00834200010020003
00834400010020003
00834600010020003
00835000010020003
00836000
00837000
00838000

```

```

">",
HDR[VENDORIDLOC] FOR VENDORIDLNG,
"<";
IF HDR[VENDORIDLOC] NEQ VENDORIDEXP THEN
BEGIN
  SHOWIT(TRUE);
  START " Expecting:";
  PUT(TXTLOC30)
  ">",
  VENDORIDEXP,
  "<";
  SHOWIT(TRUE);
  RSLT := FALSE;
END
ELSE
  IF EXPRESSMODE THEN
    SHOWIT(FALSE)
  ELSE
    SHOWIT(TRUE);

START "New Firmware Level:";
PUT(TXTLOC30)
">",
HDR[NEWFWLEVELLOC] FOR NEWFWLEVELNG,
"<";
IF EXPRESSMODE THEN
  SHOWIT(FALSE)
ELSE
  SHOWIT(TRUE);

XIT:
VALIDATEACODE := RSLT;
SHOWIT(FALSE);           * Blank line
END; * Validate Acode File
*****00869000
* Compare Product IDs of Target and Input File 0087000010010001
*****00871000
00872000
00873000010010001
00874000
00875000
00876000
00877000
00885000
00886000
00887000
00888000
00889000
00890000
00891000
00892000
00893000
00894000
00895000
00896000
00897000
00898000
00899000
00900000
00901000
00902000
00903000
00904000
00905000010010001
00906000
00907000
00908000010010001
00909000
00910000010010001

```

```

#####
# Store Firmware File Information
#####
PROCEDURE STOREFWFILEINFO;
BEGIN
  INTEGER I;
  % Assign Code file information from header record to program variables.
  REPLACE FCUSTID [0] BY HDR [CUSTIDLOC] FOR CUSTIDLNG; 00911000
  REPLACE FNEWFWLEVEL [0] BY HDR [NEWFWLEVELLOC] FOR NEWFWLEVELNG; 00912000
  REPLACE FVENDORID [0] BY HDR [VENDORIDLOC] FOR VENDORIDLNG; 00913000
  FHDRBYTES := INTEGER (HDR [HRECSIZELOC], HRECSIZELNG); 00914000
  FCODEBYTES := INTEGER (HDR [FWCODESIZELOC], FWCODESIZELNG); 00915000
  NUMDRIVEID := INTEGER (HDR [NUMDRIVEIDLOC], NUMDRIVEIDLNG); 00916000
  NUMREPFWLVL := INTEGER (HDR [NUMREPFWLVLLOC], NUMREPFWLVLNG); 00917000
  DRIVEIDLOC := NUMREPFWLVLLOC + NUMREPFWLVLNG + 00918000
  (NUMREPFWLVL * OLDFWLEVELNG); 00918200010040006
  00918400010040006
  00919000
  00920000
  00921000
  00922000
  00923000
  00924000
  00925000
  00926000
  00927000
  00928000
  00929000
  00930000
  00931000
  00932000
  00933000
  SHOW ("FROM FILE -- " C 00934000
  "CUSTID : " C FCUSTID FOR CUSTIDLNG
  ". NEW FW LEVEL : " C FNEWFWLEVEL FOR NEWFWLEVELNG,TRUE); 00935000
  SHOW ("FROM FILE -- " C 00936000
  "VENDORID : " C FVENDORID FOR VENDORIDLNG
  ". HEADERBYTES : " C FHDRBYTES FOR HRECSIZELNG DIGITS,TRUE); 00937000
  SHOW ("FROM FILE -- " C 00938000
  "CODEBYTES : " C FCODEBYTES FOR FWCODESIZELNG DIGITS C 00939000
  ". REPFWLVL : " C NUMREPFWLVL FOR NUMREPFWLVLNG DIGITS,TRUE); 00940000
  SHOW ("FROM FILE -- " C 00941000
  "NUMDRIVES : " C NUMDRIVEID FOR NUMDRIVEIDLNG DIGITS C 00942000
  ". DRIVEIDLOC: " C DRIVEIDLOC FOR DRIVEIDLNG DIGITS,TRUE); 00943000
  00944000
  END; 00945000
  00945500010020004
  IF NOT VERIFYONLY THEN
    00946000010020004
    00946200010020004
    00946400010020004
    BEGIN
      % Resize the IML buffer
      SHOW("RESIZE THE IML BUFFER.",TRUE);
      IF FCODEBYTES >= SYSCAP THEN
        BEGIN
          NROWS := FCODEBYTES DIV EIGHTK;
          REPLACE SCR[0] BY "NROWS= ", NROWS FOR * DIGITS;
          WRITE(RMT, *//, SCR);
          IF NROWS >= MAXROWS THEN
            BEGIN
              RESIZE(IMLBUF2[*,*], MAXROWS, RETAIN);
              NROWS := MAXROWS;
              NROWS := * - 1;
              FOR I := 0 STEP 1 UNTIL NROWS DO
                BEGIN
                  SHOW(" I = " C I FOR * DIGITS,TRUE);
                  RESIZE (IMLBUF2[I,*], EIGHTK, RETAIN);
                END;
              IMLP2 := IMLBUF2[0,*]; % ASSIGN POINTER
              NROWS := (FCODEBYTES - SYSCAP) DIV EIGHTK;
              NROWS := * + 1;
              RESIZE(IMLBUF3[*,*], NROWS, RETAIN);
              NROWS := * - 1;
              FOR I := 0 STEP 1 UNTIL NROWS DO
                BEGIN
                  SHOW(" I = " C I FOR * DIGITS,TRUE);
                  RESIZE (IMLBUF3[I,*], EIGHTK, RETAIN);
                END;
              IMLP3 := IMLBUF3[0,*];
            END;
        END;
    END;
  END;
  00947000
  0094750001.005.000
  0094800001.005.000
  00948500010040006
  00949000010040006
  0094902001.005.000
  0094904001.005.000
  0094906001.005.000
  0094908001.005.000
  0094920001.005.000
  0094925001.005.000
  0094930001.005.000
  0094940001.005.000
  0094950001.005.000
  0094960001.002.006
  0094980001.005.000
  0095000001.005.001
  0095010001.005.001
  0095015001.005.000
  0095020001.005.001
  0095030001.005.001
  0095040001.005.001
  0095050001.005.001
  0095055001.005.000
  0095060001.005.000
  0095070001.005.001
  0095080001.005.001
  0095090001.005.001
  00951000010040006
  00951000010040006

```

```

ELSE
  RESIZE(IMLBUF,FCODEBYTES,DISCARD);
  IMLP := IMLBUF;
  SHOW("ASSIGN POINTER TO RESIZED IMLBUF.",TRUE);
  & Build IML image from Code File

IF NOT EXPRESSMODE THEN
  SHOW("Building FW image in buffer.",TRUE);

& The LAST record of firmware code for SBC is transferred to the buffer
& with the existing NULL padding at the end of that record. The SBC
& parses out those NULLS in its internal buffer before processing the
& data.

& The LAST record of Firmware code for drives is transferred to the
& buffer with only valid data (no NULLS are included). The drives
& process the entire buffer as valid data.

IF FCODEBYTES >= 393216 THEN
BEGIN
  I := 0;
  SOFAR := 0;
  WHILE SOFAR < SYSCAP DO
BEGIN
  & REPLACE SCR[0] BY "SOFAR= ", SOFAR FOR * DIGITS;
  & WRITE(RMT,/*/,SCR);
  & DISPLAY (SCR);
  READ(CODE,CODEMRS,CODEREC);      % CODEMRS = 8192
  PC := CODEREC[0];              % ASSIGN POINTER PC TO CODEREC
  IF OPTODO = UNASSIGNCTLV THEN    % CODE IS FOR SBC
    REPLACE IMLBUF2[I,*] BY PC FOR CODEMRS
  ELSE                            % CODE IS FOR DRIVE
    BEGIN
      IF (FCODEBYTES - SOFAR) >= CODEMRS THEN % HAVE FULL DATA RECORD
        REPLACE IMLBUF2[I,*] BY PC FOR CODEMRS
      ELSE REPLACE IMLBUF2[I,*] BY PC
          FOR (FCODEBYTES - SOFAR);           % LAST
          % PARTIAL REC
    END;
  I := * + 1;
  SOFAR := * + CODEMRS;
END;
WHILE (FCODEBYTES - SOFAR) > 0 DO
BEGIN
  I := 0;
  READ(CODE,CODEMRS,CODEREC);
  PC := CODEREC[0];
  IF OPTODO = UNASSIGNCTLV THEN
    REPLACE IMLBUF3[I,*] BY PC FOR CODEMRS
  ELSE
    BEGIN
      IF (FCODEBYTES - SOFAR) >= CODEMRS THEN
        REPLACE IMLBUF3[I,*] BY PC FOR CODEMRS
      ELSE
        REPLACE IMLBUF3[I,*] BY PC FOR (FCODEBYTES - SOFAR);
    END;
  I := * + 1;
  SOFAR := * + CODEMRS;
END;
END;
ELSE
BEGIN
  SOFAR := 0;
  WHILE SOFAR < FCODEBYTES DO
BEGIN
  READ(CODE,CODEMRS,CODEREC);
  PC := CODEREC[0];
  IF OPTODO = UNASSIGNCTLV THEN    % CODE IS FOR SBC
    REPLACE IMLP:IMLP BY PC FOR CODEMRS
  ELSE                            % CODE IS FOR DRIVE
    BEGIN
      IF (FCODEBYTES - SOFAR) >= CODEMRS THEN % HAVE FULL DATA RECORD
        00951200010040006
        00951400010040006
        00951600010040006
        0095180001.005.000
        00952000
        00953000
        00955000
        0095500010020003
        00954000010020003
        00955000
        00956000
        00957000
        00958000
        00959000
        00960000
        00961000
        00962000
        00963000
        00963200010040006
        0096340001.005.000
        00963600010040006
        00964500010040006
        00965000010040006
        0096600001.005.000
        0096700001.005.000
        0096720001.005.000
        0096740001.005.000
        0096760001.005.001
        00968000010040006
        00969000010040006
        00970000
        00971000010040006
        00972000
        00973000010040006
        00974000010040006
        00975000010040006
        00976000010040006
        00976500010040006
        00977000010040006
        00977500010040006
        00978000
        00979000
        0097905001.005.000
        0097910001.005.000
        0097915001.005.000
        0097920001.005.000
        0097925001.005.000
        0097930001.005.000
        0097935001.005.000
        0097940001.005.000
        0097945001.005.000
        0097950001.005.000
        0097955001.005.000
        0097960001.005.000
        0097965001.005.000
        0097970001.005.000
        0097975001.005.000
        0097980001.005.000
        0097985001.005.000
        00980000010040006
        00980050010040006
        00980100010040006
        00980150010040006
        00980200010040006
        00980250010040006
        00980300010040006
        00980350010040006
        00980400010040006
        00980450010040006
        00980500010040006
        00980550010040006
        00980600010040006

```

```

REPLACE IMLP:IMLP BY PC FOR CODEMRS 00980650010040006
ELSE REPLACE IMLP:IMLP BY PC FOR (FCODEBYTES - SOFAR); * LAST 00980700010040006
END; * PARTIAL RECO0980750010040006
SOFAR := * + CODEMRS; 00980800010040006
END; 00980850010040006
END; 0098090001.002.004
END; * PROCEDURE STOREFWFILEINFO 00980950010040006
00981000
00982000
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00983000
* Procedure VERIFYFILE 00984000
* Procedure VERIFYFILE *0098500001.005.000
00986000
00987000
00988000
00989000
00990000
00991000
00992000
00993000
00994000
00995000
00996000
00997000
00998000
00999000
01000000
01001000
01002000
01003000
01004000
01005000
01006000
01007000
01008000
01009000
01010000
01011000
01012000
01013000
01014000
01015000
01016000
01017000
01018000
01019000
01020000
01021000
01022000
01023000
01024000
01025000
01026000
01027000
01028000
01029000
01030000
01031000
01032000
01033000
01034000
01035000
01036000
01037000
01038000
01039000
01040000
01041000
01042000
01043000
01044000

PROCEDURE VERIFYFILE;
BEGIN
  INTEGER N, X;
  LABEL GETCODE;

  * Request name of code file

  GETCODE:
  CLOSE(CODE);
  CODE.FILEUSE := VALUE(IN);
  CODE.KIND := VALUE(TAPE);
  CODE.DEPENDENTSPECS := TRUE;
  REPLACE TTL[0] BY " ";
  WHILE TTL[0] = " " DO
  BEGIN
    START
      "Enter name of code file or 'Quit'";
    PROMPT;
    IF LL > 0 THEN
    BEGIN
      IF PL = "Q" THEN
        GO GRANDXIT;
      X := LL;
      REPLACE L[LL+1] BY ".";
      SCAN PL FOR N:LL UNTIL = ".";
      REPLACE TTL[0] BY PL FOR LL UNTIL = ".", ".";
      REPLACE CODE.TITLE BY TTL[0];
      IF CODE.ATTRERR THEN
      BEGIN
        BLANK;
        SHOW("Attribute error setting CODE file title", TRUE);
        GO GETCODE;
      END;
    END; % LL > 0
  END; % WHILE

  * Attempt to locate and open code file

  * A DISK file must have 'ON <PACKNAME>' in the title declaration
  * (e.g., (UCODE)ABC/123 ON DISK), otherwise it defaults to a TAPE file.
  * If the scan ends with a "." instead of a " ", then a tape type is
  * assumed, because a disk file would require a " " for the "ON PACK"
  * part of the title declaration.

  SCAN PL:PL := TTL[0] FOR N:X UNTIL IN ENDER;
  IF PL = "." THEN % OPEN TAPE
    OPEN (CODE) % ALLOW FOR TAPE FILE EQUATE
  ELSE IF CODE.AVAILABLE = 1 THEN % DISK FILE
    CODEEOF := CODE.LASTRECORD
  ELSE % DISK FILE NOT AVAILABLE
  BEGIN
    SHOW ("The disk file name entered is NOT AVAILABLE", TRUE);
    GO GETCODE;
  END;

  CODEMRS := EIGHTK;
  IF SIZE(HDR) < CODEMRS THEN

```

```

BEGIN                                     01045000
RESIZE(HDR, CODEMRS, DISCARD);           01046000
RESIZE(CODEREC, CODEMRS, DISCARD);       01047000
END;                                       01048000
                                         01049000
READ(CODE, CODEMRS, CODEREC);           01050000
REPLACE HDR BY CODEREC FOR SIZE(HDR) WITH ASCIITOEBCDIC; 01051000
                                         01052000
% See if code file is a valid TARGET code file 01053000
                                         01054000
IF VALIDATEACODE THEN                   01055000
BEGIN                                     01056000
  IF EXPRESSMODE THEN                   01056200010020003
    SHOW("File is a valid TARGET code file",FALSE) 01056400010020003
  ELSE
    SHOW("File is a valid TARGET code file",TRUE); 01056600010020003
    SHOWIT(FALSE);                      01058000
    STOREFWFILEINFO;                   01059000
  END
ELSE
BEGIN                                     01060000
  SHOW("File is -not- a valid TARGET code file",TRUE); 01061000
  SHOWIT(FALSE);                      * Blank line 01062000
  GO GETCODE;                         01063000
END;                                       01064000
                                         01065000
                                         01066000
                                         01067000
END; % Verify File                      01068000
                                         01069000
*****01070000
% Compare Firmware Levels of Target and Input File 01071000
*****01072000
01073000
BOOLEAN PROCEDURE COMPFWLEVELS;
BEGIN                                     01074000
                                         01075000
                                         01076000
INTEGER  FWFILELOC, I;                  01077000
BOOLEAN  FWLEVELSMATCH;                01078000
                                         01079000
I := 0;                                     01080000
% WANT LOWER FOUR BYTES (3,2,1,0) OF 8 BYTE FIELD 01081000
                                         01082000
FWFILELOC := NUMREPFWLVL$LOC; % OFFSET BY '4' FOR PROPER LOOPING 01083000
                                         % INCREMENT BELOW. 01084000
WHILE (((I:= *+1) LEQ NUMREPFWLVL$) AND (NOT FWLEVELSMATCH)) DO 01085000
BEGIN                                     01086000
  FWLEVELSMATCH := TPRODREVLVL = HDR [{(FWFILELOC:= *+ OLDFWLEVELNG)}] 01087000
                                         FOR NEWFWLEVELNG; 01088000
  IF DEBUG THEN                         % SHOW ENTIRE 7 OR 8 BYTES OF FILE ENTRY 01089000
                                         % IF NULLS ARE PRINTED, OUTPUT IS LOST. 01089100010010001
  IF OPTODO = UNASSIGNUNITV THEN        01089200010010001
  BEGIN
    IF HDR [(FWFILELOC - 4)] = 48"00" THEN % BYTE [0] = NULL 01089400010010001
      SHOW ("FILE FW-LOOP = >" C HDR [(FWFILELOC - 3)] 01091000
                                         FOR (OLDFWLEVELNG - 1) C "<", TRUE) 01092000
    ELSE      % BYTE [0] NEQ NULL. FORMAT REQUIRED FOR SCSI DRIVES 01093000
      SHOW ("FILE FW-LOOP = >" C HDR [(FWFILELOC - 4)] 01094000
                                         FOR OLDFWLEVELNG C "<", TRUE); 01095000
  END
  ELSE      % FOR SBC 01095200010010001
    SHOW ("FILE FW-LOOP = >" C HDR [FWFILELOC] FOR NEWFWLEVELNG 01095400010010001
                                         C "<", TRUE); 01095600010010001
  END;                                     01095800010010001
                                         01096000
                                         01097000
IF NOT FWLEVELSMATCH THEN      % NO MATCH FOUND 01098000
BEGIN                                     01099000
  SHOW ("<< Firmware level of target is NOT in input file. >>",TRUE); 01100000
  SHOW ("<< Download will NOT take place. >>", TRUE); 01101000
  COMPFWLEVELS := FALSE;                 01102000
  END
ELSE
BEGIN                                     01103000
                                         01104000
                                         01105000

```

```

IF OPTODO = UNASSIGNUNITV THEN      % Drives only. Get full 8 bytes 01106000
  BEGIN
    REPLACE SAVEFWLVL [0] BY HDR [(FWFILELOC - 4)] FOR OLDFWLEVELNG; 01108000
    IF DEBUG THEN
      SHOW ("SAVEFWLVL (FORMAT(IVR) NEEDED?-BYTE[0] NEQ 48'00') = >" C01110000010020004
        SAVEFWLVL FOR OLDFWLEVELNG C "<", TRUE); 01111000
    END;
    COMPFWLEVELS := TRUE;
  END; 01112000
  01113000
  01114000
  01115000
  01116000
END; % PROCEDURE COMPFWLEVELS 0111601001.005.000
%%%%% 0111602001.005.000
% Compare SERVO Firmware Levels of Target and Input File 0111603001.005.000
%%%%% 0111604001.005.000
BOOLEAN PROCEDURE COMPSERVOFWLVL;
BEGIN 0111605001.005.000
  0111606001.005.000
  0111607001.005.000
  0111608001.005.000
  0111609001.005.000
  0111610001.005.000
  0111611001.005.000
  0111612001.005.000
  0111613001.005.000
  FWFILELOC := NUMREPEFWLVL; % OFFSET BY '4' FOR PROPER LOOPING 0111614001.005.000
  % INCREMENT BELOW. 0111615001.005.000
  WHILE (((I:= *+1) LEQ NUMREPEFWLVL) AND (NOT FWLEVELSMATCH)) DO 0111616001.005.000
    BEGIN
      FWLEVELSMATCH := TSERVOREVVLV = HDR [(FWFILELOC:= *+ OLDFWLEVELNG)] 0111617001.005.000
        FOR NEWFWLEVELNG; 0111618001.005.000
        0111619001.005.000
    % SHOW("SERVO FIRMWARE =>" C 0111619201.005.000
    % TSERVOREVVLV FOR NEWFWLEVELNG, TRUE); 0111619401.005.000
    IF DEBUG THEN
      % SHOW ENTIRE 7 OR 8 BYTES OF FILE ENTRY 0111620001.005.000
      % IF NULLS ARE PRINTED, OUTPUT IS LOST. 0111621001.005.000
      IF OPTODO = UNASSIGNUNITV THEN 0111624001.005.000
        BEGIN
          IF HDR [(FWFILELOC - 4)] = 48"00" THEN % BYTE [0] = NULL 0111625001.005.000
            SHOW ("FILE FW-LOOP = >" C HDR [(FWFILELOC - 3)] 0111626001.005.000
              FOR (OLDFWLEVELNG - 1) C "<", TRUE) 0111627001.005.000
            0111628001.005.000
          ELSE % BYTE [0] NEQ NULL. FORMAT REQUIRED FOR SCSI DRIVES 0111629001.005.000
            SHOW ("FILE FW-LOOP1 = >" C HDR [(FWFILELOC - 4)] 0111630001.005.000
              FOR OLDFWLEVELNG C "<", TRUE); 0111631001.005.000
            0111632001.005.000
          END
        ELSE % FOR SBC 0111633001.005.000
          SHOW ("FILE FW-LOOP2 = >" C HDR [FWFILELOC] FOR NEWFWLEVELNG 0111634001.005.000
            C "<", TRUE); 0111635001.005.000
        END;
      COMPSERVOFWLVL := TRUE;
    END; 0111636001.005.000
    0111637001.005.000
    0111653001.005.000
  END; % PROCEDURE COMPFWLEVELS 0111655001.005.000
  $ PAGE 0111656001.005.000
  01117000
  01118000
%%%%% 01119000
% Release the target 01120000
%%%%% 01121000
  01122000
BOOLEAN PROCEDURE RELEASETARGET (OPTODO); 01123000
VALUE OPTODO; 01124000
INTEGER OPTODO; 01125000
BEGIN 01126000
  01127000
  INTEGER N, RSLT;
  01128000
  01129000
% Release the target 01130000
  RSLT := USERMAINTREQUEST(CTLUNIT,OPTODO,0,0,0,
    SHORTBUF,MRD,HDPRESULT);
  01131000
  IF RSLT > 0 THEN 01132000
    BEGIN
      SHOW("Unable to release TARGET " C 01133000
        01134000
        01135000
        01136000
    END
  END

```

```

        CTLUNIT FOR * DIGITS          C          01137000
        ". Error"                   C          01138000
        RSLT FOR * DIGITS,TRUE);    C          01139000
        RELEASESETARGET := FALSE;
        END
ELSE
BEGIN
        RELEASESETARGET := TRUE;
        IF OPTODO = UNASSIGNUNITV THEN
        BEGIN
        SDMASSIGNED := FALSE;
        SHOWRSLT (RSLT, UNASSIGNUNITV);
        END
ELSE
BEGIN
        CTCLASSIGNED := FALSE;
        SHOWRSLT (RSLT, UNASSIGNNCTLV);
        END;
END;
END; % BOOLEAN PROCEDURE RELEASESETARGET
***** Load SBC Firmware Procedure *****01158000
% Load SBC Firmware Procedure          01159000
*****01160000
01161000
PROCEDURE LOADCTLFW;
BEGIN
        INTEGER N,RSLT,W,OFFSET,I;
        LABEL NEXTCTL;
        ARRAY T[0:2];
        % System call to determine machine type
        % If we are not running on an IOM type system, terminate.
        SYSTEMTYPE := TIME(24);
        REPLACE T BY SYSTEMTYPE FOR 6 ;
        IF T ISNT "A11" FOR 3 AND
        T ISNT "A14" FOR 3 AND
        T ISNT "A12" FOR 3 AND
        T ISNT "A16" FOR 3 AND
        T ISNT "A18" FOR 3 AND
        T ISNT "A19" FOR 3 AND
        T ISNT "A28" FOR 3 AND
        T ISNT "NX46" FOR 4 AND
        T ISNT "NX48" FOR 4 AND
        T ISNT "NX58" FOR 4 THEN
        BEGIN
        SHOW("SYSTEMTYPE = " C SYSTEMTYPE FOR 6, TRUE);
        SHOW("Running system is not an IOM system. Program is ENDING",TRUE);01175000010010001
        GO GRANDXIT;
        END;
        SHOW("SYSTEMTYPE = " C SYSTEMTYPE FOR 6, TRUE);
        START
        "Caution: download FW to only one CTL in a string at a time";
        SHOWIT(TRUE);
        START "          CTL must be URed to all visible hosts";
        SHOWIT(TRUE);
        START "          Other CTL must be online";
        SHOWIT(TRUE);
        ***** MAIN TARGET (CTL) LOOP *****01187000
        OPTODO := UNASSIGNNCTLV;      % CTL PARAMETER TO UMR
        WHILE TRUE DO
        BEGIN
        %Request CTL number
        CTLUNIT := 0;
        WHILE CTLUNIT = 0 DO
        BEGIN

```



```

SHOWATTRIBUTES;

% Get code file & match header rec. info against Target info.

IF TVENDORID NEQ VENDORIDEXP FOR VENDORIDLNG THEN
  BEGIN
    SHOW ("Vendor ID of target is not UNISYS.", TRUE);
    SHOW ("<> Download will NOT take place. >>", TRUE);
    IF NOT RELEASETARGET (OPTODO) THEN
      GO GRANDKIT
    ELSE
      GO NEXTCTL;
  END;
IF NOT COMPPRODIDS THEN  % COMPARE PRODUCT IDs OF FILE VS. TARGET
  IF NOT RELEASETARGET (OPTODO) THEN
    GO GRANDKIT
  ELSE
    GO NEXTCTL;
IF NOT COMPFWLEVELS THEN  % COMPARE FIRMWARE LEVELS OF FILE VS. TARGET
  IF NOT RELEASETARGET (OPTODO) THEN
    GO GRANDKIT
  ELSE
    GO NEXTCTL;
IF TPRODREVlvl = FNEWFWLEVEL FOR NEWFWLEVELNG THEN
  BEGIN
    SHOW ("Firmware levels of Target and File are the same.",TRUE);
    START "Do you still want to download the firmware? Enter YES or NO";
    PROMPT;
    IF PL NEQ "Y" THEN
      BEGIN
        SHOW ("Download will not take place for target " C
              CTLUNIT FOR * DIGITS, TRUE);
        IF NOT RELEASETARGET (OPTODO) THEN
          GO GRANDKIT
        ELSE
          GO NEXTCTL;
      END;
  END;
% Request function

SHOW("Starting to download code to CTL " C
      CTLUNIT FOR * DIGITS,TRUE);
IF SYSCAP <= FCODEBYTES THEN
  BEGIN
    NUMBROFIOS := FCODEBYTES DIV EIGHTK;
    SIZEOFLSTIO := FCODEBYTES MOD EIGHTK;
    OFFSET := 0;
    I := 0;
    DO
      BEGIN
        RSLT := USERMAINTRREQUEST(CTLUNIT, DOWNLOADMODE6, 8192, OFFSET, 0,
                                   IMLBUF2[I,*],MRD,HDPRESULT);
        IF RSLT > 0 THEN
          BEGIN
            SHOWRSLT(RSLT,LOADSLAVEIMLV);
            SHOWMRDBITS;
            SHOWHDPRESULT;

            SHOW("<> Microcode NOT loaded!! >>",TRUE);
            IF NOT RELEASETARGET (OPTODO) THEN
              GO GRANDKIT;
          END;
        REPLACE SCR[0] BY "I= ", I FOR *DIGITS;
        DISPLAY (SCR);
        REPLACE SCR2[0] BY "NUMBROFIOS= ", NUMBROFIOS FOR * DIGITS;
        DISPLAY (SCR2);
        NUMBROFIOS := * - 1;
      END;
  END;

```

```

        OFFSET := * + EIGHTK;
        I := * + 1;
    END
    UNTIL NUMBROFIOS = 1;
    RSLT := USERMAINTREQUEST(CTLUNIT, DOWNLOADMODE7, 8192, OFFSET, 0,
                             IMLBUF2[I,*], MRD, HDPRESULT);
END
ELSE
RSLT := USERMAINTREQUEST(CTLUNIT, LOADSLAVEIMLV, FCODEBYTES, 0, 0,
                         IMLBUF, MRD, HDPRESULT);
IF RSLT > 0 THEN
BEGIN
    SHOWRSLT(RSLT, LOADSLAVEIMLV);
    SHOWMRDBITS;;
    SHOWHDPRESULT;

    SHOW("<< Microcode NOT loaded!! >>", TRUE);
    IF NOT RELEASESETARGET (OPTODO) THEN
        GO GRANDXIT
    ELSE
        GO NEXTCTL;
END;

SHOW("Download complete. Waiting to read attributes of CTL " C
     CTLUNIT FOR * DIGITS, TRUE);
SHOW("Do not power off or alter CTL " C
     CTLUNIT FOR * DIGITS, TRUE);

* The SBC has now turned off its SCSI interface.
* It won't come alive until after it has done the power up
* confidence tests (approx. 30 seconds).
* Initially wait 30 seconds and then cycle ATTRIBUTES calls
* every 10 seconds until it comes back to life or drops dead.

WHEN(30);
SHOW("00:30 Starting to read attributes of CTL " C
     CTLUNIT FOR * DIGITS, TRUE);
TIMER := 0;
REPLACE SHORTBUF [0] BY NUL FOR SIZE (SHORTBUF);

RSLT := USERMAINTREQUEST(CTLUNIT, TESTUNITREADY, 0, 0, 0,
                         SHORTBUF, MRD, HDPRESULT);

IF RSLT > 0 THEN
BEGIN
    SHOWRSLT(RSLT, LOADSLAVEIMLV);
    SHOWMRDBITS;
    SHOWHDPRESULT;

    SHOW("<< Test Unit Ready Failed!! >>", TRUE);
    IF NOT RELEASESETARGET (OPTODO) THEN
        GO GRANDXIT;
END;
DISPLAY(" Test Unit Ready - OK ");

DO BEGIN
WHEN(10);
RSLT := USERMAINTREQUEST(CTLUNIT, ATTRIBUTESV, 254, 0, 0,
                         SHORTBUF, MRD, HDPRESULT);
N := (TIMER:=*+1)*10 + 30;           * Seconds
SHOWRSLT(RSLT, -N);

END UNTIL N >= 1*60                * Drop dead time
        OR RSLT = 0;                  * Success

IF RSLT > 0 THEN                  * Timeout
BEGIN
    SHOW("MCP interface error " C
         RSLT FOR * DIGITS      C
         " after FW download" C
         " timeout on CTL "   C

```

```

        CTLUNIT FOR * DIGITS,TRUE);
        SHOWRSLT(RSLT,ATTRIBUTESV);
        SHOW("Check CTL for malfunction",TRUE);
        GO GRANDXIT;
END;

SHOW("CTL"           C
      CTLUNIT FOR * DIGITS C
      " completed FW download",TRUE);

* Format attributes

SHOWATTRIBUTES;

SHOW ("                                     ", TRUE);
SHOW ("If BOOT code was loaded instead of OPERATIONAL code", TRUE);
SHOW ("The firmware level of the SBC may not have changed ", TRUE);
SHOW ("                                     ", TRUE);

* Release the CTL

IF NOT RELEASESETARGET (OPTODO) THEN
  GO GRANDXIT;

SHOW("Okay to UR- CTL"           C
      CTLUNIT FOR * DIGITS C
      " and load companion CTL",TRUE);

NEXTCTL:
END;  * WHILE TRUE DO

END; * Load CTL Firmware

#####
# Load SCSI Drive Firmware Procedure
#####

PROCEDURE LOADEDEVW;
BEGIN
  INTEGER N,RSLT,W,OFFSET,I,J;
  LABEL NEXTDRIVE;
  ARRAY T[0:2];
* System call to determine machine type
* If we are not running on an IOM type system, terminate.

  SYSTEMTYPE := TIME(24);
* REPLACE T BY SYSTEMTYPE FOR 3 ;
* IF T ISNT "A11" FOR 3 AND
*   T ISNT "A14" FOR 3 AND
*   T ISNT "A16" FOR 3 AND
*   T ISNT "A17" FOR 3 AND
*   T ISNT "A18" FOR 3 AND
*   T ISNT "A19" FOR 3 AND
*   T ISNT "A28" FOR 3 THEN
BEGIN
  SHOW("SYSTEMTYPE = " C SYSTEMTYPE FOR 6, TRUE);
  SHOW("Running system does not support this drives." C
        " Program is ENDING.",TRUE);
  GO GRANDXIT;
END;
SHOW("SYSTEMTYPE = " C SYSTEMTYPE FOR 6,TRUE);
IF NOT EXPRESSMODE THEN
  BEGIN
    START "Caution: Load only one drive in a string at a time";
    SHOWIT(TRUE);
    START "      drive must be URed to all visible hosts";
    SHOWIT(TRUE);
    START "      If using Fibre Channel UR the opposite Port";
    SHOWIT(TRUE);
  END;

```

```

01336000
01337000
01338000
01339000
01340000
01341000
01342000
01343000
01344000
01348000
01349000
01350000
01351000
01351100010010001
01351200010010001
01351300010010001
01351400010010001
01351500010010001
01352000
01353000
01354000
01355000
01356000
01357000
01357200010020002
01357400010020002
01357600010020002
01357800010020002
01358000
01359000
01360000
01361000
01362000
01363000
01364000
0136500001.005.001
01366000
01367000
0136800001.005.001
01369000
0137000001.005.001
01371000
01372000010030005
01373000
01374000
01375000
01376000010030005
01376500010030005
01377000010030005
01377500010030005
01378000010030005
01379000010030005
01379500010030005
01380000010030005
01380500010030005
01381000010030005
01381500010030005
01382000010030005
01383000010030005
01384000010030005
01385000010030005
01386000010030005
01386200010020003
01386400010020003
01387000010020003
01388000010020003
01389000010020003
01390000010020003
0139020001.005.000
0139040001.005.000
01390500010020003

```



```

IF RSLT > 0 THEN          * Problem          01433000
BEGIN
  SHOW("MCP interface error"      C          01434000
    RSLT FOR * DIGITS          C          01435000
    " to get target"          C          01436000
    " attributes for SCSI drive" C          01437000
    CTLUNIT FOR * DIGITS,TRUE);
  SHOWRSLT(RSLT,ATTRIBUTESV);
  SHOW("Check TARGET for problem",TRUE);
  IF NOT RELEASESETARGET (OPTODO) THEN
    GO GRANDXIT
  ELSE
    GO NEXTDRIVE;
END;

SHOWATTRIBUTES;

IF EXPRESSMODE THEN      * Save for display at end of download 01449200010020003
  REPLACE OLDFWLEVEL [0] BY TPRODREVLVL FOR NEWFWLEVELNG;
 01449400010020003
 01449600010020003

* Get code file & match header rec. info against Target info.

IF TVENDORID NEQ VENDORIDEXP FOR VENDORIDLNG THEN          01450000
BEGIN
  SHOW ("Vendor ID of target is not UNISYS.", TRUE);          01451000
  SHOW ("<< Download will NOT take place. >>", TRUE);          01452000
  IF NOT RELEASESETARGET (OPTODO) THEN          01453000
    GO GRANDXIT
  ELSE
    GO NEXTDRIVE;
END;
IF NOT COMPPRODIDS THEN  * COMPARE PRODUCT IDs OF FILE vs. TARGET 01454000
  IF NOT RELEASESETARGET (OPTODO) THEN          01455000
    GO GRANDXIT
  ELSE
    GO NEXTDRIVE;
 01456000
 01457000
 01458000
 01459000
 01460000
 01461000
 01462000010010001
 01463000
 01464000
 01465000
 01466000
 01467000
 01468000
 01469000
 01470000
 01471000
 01472000
 01473000
 0147400001.005.000
 01475000
 01476000
 01477000
 01478000
 01479000
 01480000
 01481000
 01482000
 01483000
 01484000
 01485000
 01486000
 01487000
 01487500010010001
 01488000
 01489000
 0148920001.005.000
 01489400010010001
 01489600010010001
 01489800010010001
 01490000010010001
 01490200010010001
 01491000010010001
 01492000010010001
 01493000010010001
 01494000010010001
 01495000010010001
 01496000010010001
 01497000010010001
 01498000
 01499000
 01500000
 01502000

```

```

* Get attributes of Target
RSLT := USERMAINTREQUEST(CTLUNIT,INQUIRY,23,192,0,
                           SHORTBUF,MRD,HDPRESULT);
IF RSLT > 0 THEN          * Problem
BEGIN
  SHOW("MCP interface error"      C
       RSLT FOR * DIGITS      C
       " to get target"      C
       " attributes for SCSI drive" C
       CTLUNIT FOR * DIGITS,TRUE);
  SHOWRSLT(RSLT,ATTRIBUTESV);
  SHOW("Check TARGET for problem",TRUE);
  IF NOT RELEASESETARGET (OPTODO) THEN
    GO GRANDXIT
  ELSE
    GO NEXTDRIVE;
END;
SHOWINQUIRYDATA;

IF EXPRESSMODE THEN      * Save for display at end of download
  REPLACE OLDFWLEVEL [0] BY TSERVOREVLVL FOR NEWFWLEVELNG;

* Get code file & match header rec. info against Target info.

IF TSERVOREVLVL = FNEWFWLEVEL FOR NEWFWLEVELNG THEN
BEGIN
  SHOW ("Servo FW levels of Target and File are the same.",TRUE);
  START "Do you still want to download the firmware? Enter YES or NO";0150245001.005.001
  PROMPT;
  IF PL NEQ "Y" THEN
    BEGIN
      SHOW ("Download will not take place for target" C
            CTLUNIT FOR * DIGITS, TRUE);
      IF NOT RELEASESETARGET (OPTODO) THEN
        GO GRANDXIT
      ELSE
        GO NEXTDRIVE;
    END
  ELSE REPLACE SAVEFWLVL [0] BY 48"00" FOR OLDFWLEVELNG;  * NO FORMAT0150256001.005.001
END
ELSE
IF NOT COMPSERVOFWLVL THEN  * COMPARE FW LEVELS OF FILE VS. TARGET 0150259001.005.000
  IF NOT RELEASESETARGET (OPTODO) THEN
    GO GRANDXIT
  ELSE
    GO NEXTDRIVE;

* If drive needs to be formatted after firmware download, get
* permission before downloading the code.
* Each FW level area in file = 8 bytes. If byte 0 neq 48"00", drive
* must be formatted after code is loaded.
IF SAVEFWLVL [0] NEQ 48"00" THEN  * Drive needs formatting
  IF NOT OKTOFORMAT THEN
    IF NOT RELEASESETARGET (OPTODO) THEN
      GO GRANDXIT
    ELSE
      GO NEXTDRIVE;

* Request function

IF NOT EXPRESSMODE THEN
  SHOW("Starting to download code to drive" C
       CTLUNIT FOR * DIGITS,TRUE);
IF FCODEBYTES > SYSCAP THEN
BEGIN
  NUMBROFIOS := FCODEBYTES DIV EIGHTK;
  NUMBROFIOS := * + 1;
  SHOW(" NUMBROFIOS = " C NUMBROFIOS FOR * DIGITS,TRUE);

```

DECISION  
ON  
SYSTEM  
BYTES

```

SIZEOFLSTIO := FCODEBYTES MOD EIGHTK;
SHOW(" SIZEOFLSTIO = " C SIZEOFLSTIO FOR * DIGITS,TRUE);
FIRSTTIME := TRUE;
OFFSET := 0;
I := 0;
J := 0;
DO
BEGIN
  IF J <= 47 THEN
  BEGIN
    RSLT := USERMAINTREQUEST(CTLUNIT, DOWNLOADMODE7, 8192, OFFSET, 0,
      IMLBUF2[I,*], MRD, HDPRESULT);
    SHOW("1I = " C I FOR * DIGITS,TRUE);
    SHOW("1OFFSET = " C OFFSET FOR * DIGITS,TRUE);
    IF RSLT > 0 THEN
      BEGIN
        SHOWRSLT(RSLT, LOADSLAVEIMLV);
        SHOWMRDBITS;
        SHOWHDPRESULT;

        SHOW("<< Microcode(1) NOT loaded!! >>",TRUE);
        IF NOT RELEASESETARGET (OPTODO) THEN
          GO GRANDXIT;
      END;
    END;
  ELSE
  BEGIN
    IF FIRSTTIME THEN
      I := 0;
    FIRSTTIME := FALSE;
    RSLT := USERMAINTREQUEST(CTLUNIT, DOWNLOADMODE7, 8192, OFFSET, 0,
      IMLBUF3[I,*], MRD, HDPRESULT);
    SHOW("2I = " C I FOR * DIGITS,TRUE);
    SHOW("2OFFSET = " C OFFSET FOR * DIGITS,TRUE);
    SHOW("2J = " C J FOR * DIGITS,TRUE);
    IF RSLT > 0 THEN
      BEGIN
        SHOWRSLT(RSLT, LOADSLAVEIMLV);
        SHOWMRDBITS;
        SHOWHDPRESULT;

        SHOW("<< Microcode(2) NOT loaded!! >>",TRUE);
        IF NOT RELEASESETARGET (OPTODO) THEN
          GO GRANDXIT;
      END;
    END;
  END;
  NUMBROFIOS := * - 1;
  OFFSET := * + EIGHTK;
  I := * + 1;
  J := * + 1;
END
UNTIL NUMBROFIOS = 0;
SHOW(" NUMBROFIOS = " C NUMBROFIOS FOR * DIGITS,TRUE);
SHOW(" I = " C I FOR * DIGITS,TRUE);
SHOW(" J = " C J FOR * DIGITS,TRUE);
SHOW(" OFFSET = " C OFFSET FOR * DIGITS,TRUE);
IF J < 47 THEN
BEGIN
  RSLT := USERMAINTREQUEST(CTLUNIT, DOWNLOADMODE7, 8192, OFFSET, 0,
    IMLBUF2[I,*], MRD, HDPRESULT);
  SHOW("3I = " C I FOR * DIGITS,TRUE);
  SHOW("3OFFSET = " C OFFSET FOR * DIGITS,TRUE);
  IF RSLT > 0 THEN
    BEGIN
      SHOWRSLT(RSLT, LOADSLAVEIMLV);
      SHOWMRDBITS;
      SHOWHDPRESULT;

      SHOW("<< Microcode(3) NOT loaded!! >>",TRUE);
      IF NOT RELEASESETARGET (OPTODO) THEN
        GO GRANDXIT;
    END;
END;

```

```

01506400010040006
0150640201.005.000
0150640401.005.000
01506410010040006
01506415010040006
0150641601.005.000
01506420010040006
01506430010040006
0150643501.005.000
0150643601.005.000
01506440010040006
0150645001.005.000
0150645201.005.000
0150645401.005.000
0150645601.005.000
0150645801.005.000
0150646001.005.000
0150646201.005.000
0150646401.005.000
0150646601.005.000
0150646801.005.000
0150647001.005.000
0150647201.005.000
0150647401.005.000
0150647601.005.000
0150647801.005.000
0150648001.005.000
0150648201.005.000
0150648401.005.000
0150648601.005.000
0150648801.005.000
0150649001.005.000
0150649201.005.000
0150649401.005.000
0150649601.005.000
0150649801.002.006
0150650001.002.006
0150650201.002.006
0150650401.002.006
0150650601.002.006
0150650801.002.006
0150651001.005.000
0150651201.002.006
0150651401.002.006
0150651601.002.006
0150651801.005.000
0150652001.002.006
0150652201.002.006
0150652401.002.006
0150652601.005.000
0150652801.002.006
0150653001.005.000
0150653201.005.000
0150653401.005.000
0150653601.005.000
0150653801.005.000
0150654001.005.000
0150654201.005.000
0150654401.005.000
0150655001.005.000
0150655201.005.000
0150655401.005.000
0150655601.005.000
0150655801.005.000
0150656001.005.000
0150656201.005.000
0150656401.005.000
0150656601.005.000
0150656801.005.000
0150657001.005.000
0150657201.005.000

```

393 216 Bytes

Choose 1st Two-Dimens  
Array

Choose 2nd Two  
Dimens. Array

```

        END;
    END
ELSE
BEGIN
    RSLT := USERMAINTREQUEST(CTLUNIT, DOWNLOADMODE7, 4224, OFFSET, 0,
        IMLBUF3[I,*], MRD, HDRESULT);
    SHOW("4I = " C I FOR * DIGITS, TRUE);
    SHOW("4OFFSET = " C OFFSET FOR * DIGITS, TRUE);
    SHOW("4J = " C J FOR * DIGITS, TRUE);
    IF RSLT > 0 THEN
        BEGIN
            SHOWRSLT(RSLT, LOADSLAVEIMLV);
            SHOWMRDBITS;
            SHOWHDPRESULT;

            SHOW("<< Microcode(4) NOT loaded!! >>", TRUE);
            IF NOT RELEASESETARGET (OPTODO) THEN
                GO GRANDXIT;
        END;
    END;
ELSE
BEGIN
    RSLT := USERMAINTREQUEST(CTLUNIT, LOADSLAVEIMLV, FCODEBYTES, 0, 0,
        IMLBUF, MRD, HDRESULT);
    SHOW("5I = " C I FOR * DIGITS, TRUE);
    SHOW("5OFFSET = " C OFFSET FOR * DIGITS, TRUE);
    IF RSLT > 0 THEN
        BEGIN
            SHOWRSLT(RSLT, LOADSLAVEIMLV);
            SHOWMRDBITS;;
            SHOWHDPRESULT;

            SHOW("<< Microcode(5) NOT loaded!! >>", TRUE);
            IF NOT RELEASESETARGET (OPTODO) THEN
                GO GRANDXIT;
            ELSE
                GO NEXTDRIVE;
        END;
    END;
    IF EXPRESSMODE THEN
        SHOW("Download complete. Waiting 20 seconds for prom burn to " C
            CTLUNIT FOR * DIGITS, TRUE)
    ELSE
        SHOW("Download complete. Waiting 60 seconds for prom burn to " C
            CTLUNIT FOR * DIGITS, TRUE);
    SHOW("Do not power off or alter drive " C
        CTLUNIT FOR * DIGITS, TRUE);

    * The SCSI disk drive has now turned off its SCSI interface.
    * It won't come alive until after it has done the power up
    * confidence tests (approx. 60 seconds). The second ATTRIBUTES command
    * can then be issued to display the new firmware level. NOTE: If the
    * second ATTRIBUTES command is issued before the target sequences
    * (powers itself back up) the target will hang and the program will show
    * an error.

    IF NOT EXPRESSMODE THEN
        SHOW("00:10 - Waiting for prom burn to target " C
            CTLUNIT FOR * DIGITS, TRUE);
    TIMER := 0;

    IF NOT EXPRESSMODE THEN
        DO BEGIN
            WHEN(10);
            N := (TIMER:=+1)*10 + 10;           * Seconds
            IF NOT EXPRESSMODE THEN
                SHOWRSLT(RSLT, -N);
        END UNTIL N >= 1*60;                * Drop dead time
    IF EXPRESSMODE THEN
        WHEN (20);

```

D 30

LOAD SINGLE  
ARRAY

```

REPLACE SHORTBUF [0] BY NUL FOR SIZE (SHORTBUF); 01543620010040006
RSLT := USERMAINTREQUEST(CTLUNIT,TESTUNITREADY,0,0,0, 01543640010040006
                           SHORTBUF,MRD,HDPRESULT); 01543660010040006
IF RSLT > 0 THEN 01543680010040006
  BEGIN 01543700010040006
    SHOWRSLT(RSLT,LOADSLAVEIMLV); 01543720010040006
    SHOWMRDBITS; 01543740010040006
    SHOWHDPRESULT; 01543760010040006
    SHOW("<< Test Unit Ready Failed!! >>",TRUE); 01543780010040006
    IF NOT RELEASESETARGET (OPTODO) THEN 01543800010040006
      GO GRANDXIT; 01543820010040006
    END; 01543840010040006
  DISPLAY(" Test Unit Ready - OK "); 01543860010040006
  REPLACE SHORTBUF [0] BY NUL FOR SIZE (SHORTBUF); 01543880010040006
RSLT := USERMAINTREQUEST(CTLUNIT,ATTRIBUTESV,254,0,0, 01543900010040006
                           SHORTBUF,MRD,HDPRESULT); 01544000010040006
IF RSLT > 0 THEN          * Timeout 01544500010040006
  BEGIN 01545000
    SHOW("MCP interface error "      C 01546000
         RSLT FOR * DIGITS          C 01547000
         " after FW download"     C 01548000
         " timeout on drive "     C 01549000
         CTLUNIT FOR * DIGITS,TRUE); 01550000
    SHOWRSLT(RSLT,ATTRIBUTESV); 01551000010020003
    SHOW("Check the drive for malfunction",TRUE); 01552000
    IF EXPRESSMODE THEN 01553000
      SHOW(">> Run DFAST again NOT using Express Mode <<",TRUE); 01554000
      GO GRANDXIT; 01555000
    END; 01556000
    SHOW("Target " C CTLUNIT FOR * DIGITS C " completed download",TRUE); 01557000
    SHOWATTRIBUTES; 01558000
  * Get attributes of Target 01558200010020003
  RSLT := USERMAINTREQUEST(CTLUNIT,INQUIRY,23,192,0, 01558400010020003
                           SHORTBUF,MRD,HDPRESULT); 01559000
  IF RSLT > 0 THEN          * Problem 01560000010020003
    BEGIN 01561000
      SHOW("MCP interface error "      C 01563000010020003
           RSLT FOR * DIGITS          C 01565000
           " to get target"        C 01566000
           " attributes for SCSI drive " C 01571000
           CTLUNIT FOR * DIGITS,TRUE); 0157100201.005.000
      SHOWRSLT(RSLT,ATTRIBUTESV); 0157100401.005.000
      SHOW("Check TARGET for problem",TRUE); 0157100601.005.000
      IF NOT RELEASESETARGET (OPTODO) THEN 0157100801.005.000
        GO GRANDXIT; 0157101001.005.000
      ELSE 0157101201.005.000
        GO NEXTDRIVE; 0157101401.005.000
    END; 0157101601.005.000
    SHOWINQUIRYDATA; 0157101801.005.000
  IF EXPRESSMODE THEN 0157102001.005.000
    SHOW("ProdID-" C TPRODID FOR DRIVEIDLNG C "OldFWlvl-" C 0157102201.005.000
         OLDFWLEVEL [0] FOR NEWFWLEVELNG   C ".NewFWlvl-" C 0157102401.005.000
         TPRODREVLVL   FOR NEWFWLEVELNG,TRUE); 0157102601.005.000
  * Release the target drive 0157102801.005.000
  IF NOT RELEASESETARGET (OPTODO) THEN 0157103001.005.000
    GO GRANDXIT; 0157103201.005.000
                                            0157103401.005.000
                                            0157103601.005.000
                                            0157103801.005.000
                                            0157104001.005.000
                                            0157104201.005.000
                                            01571100010020003
                                            01571200010020003
                                            01571300010020003
                                            01571400010020003
                                            01571500010020003
                                            01572000
                                            01573000
                                            01574000
                                            01575000
                                            01575200010020002

```

```

SHOW("Okay to UR- target "      C          01575400010020002
      CTLUNIT FOR * DIGITS      C          01575600010020002
      " and select another drive.",TRUE); 01575800010020002
                                         01576000
NEXTDRIVE:                           01577000
  END;  & WHILE TRUE DO
                                         01578000
END; & Load SCSI drive Firmware 01579000
$$ PAGE
                                         01580000
                                         01581000
                                         01582000
*****01583000
%      Initialization Block 01584000
*****01585000
                                         01586000
PROCEDURE INITIALIZE; 01587000
BEGIN
                                         01588000
   INTEGER N; 01589000
                                         01590000
   ODTMODE   := MYSELF.INITIATOR >= 0; 01591000
   ATODT    := MYSELF.INITIATOR  = 0; 01592000
                                         01593000
   SDMASSIGNED := 01594000
   CTLASSIGNED := FALSE; 01595000
                                         01596000
                                         01597000
   TSTMP[0] := COMPILETIME(15);      % mmddyy 01598000
   TSTMP[1] := COMPILETIME(09);      % hhmmss 01599000
   TSTMP[2] := TIME(15);           % mmddyy 01600000
   TSTMP[3] := TIME(09);           % hhmmss 01601000
                                         01602000
* See if we are privileged and can make use of the MCP interface. 01603000
                                         01604000
N := LINKLIBRARY(MCP); 01605000
IF DEBUG THEN
BEGIN
  START "MCP linkage result = ", 01606000
        N FOR * NUMERIC;
  IF N = 1 THEN
    ADD " (normal)";
    SHOWIT(TRUE);
  END;
  IF N < 0 THEN
BEGIN
  SHOW("FW download not possible. Program is ENDING.",TRUE); 01616000010010001
  GO GRANDXIT;
END;
  BLANK;
  START
    MYSELF.NAME,
    " (Version ",
    COMPILETIME(20) FOR 2 DIGITS,
    ".",
    COMPILETIME(21) FOR 3 DIGITS,
    ") compiled ",
    ETSTMP[00] WITH DATETIME,
    ", run on ",
    ETSTMP[12] WITH DATETIME;
  SHOWIT(TRUE);
  SHOWIT(FALSE);  % Blank line 01631000
                                         01632000
                                         01633000
  START
  "DO YOU WANT TO VIEW THE USER DOCUMENTATION FOR DFAST? 'Y' OR 'N'"; 01634000
  PROMPT;
  IF PL = "Y" THEN
    USERPROCESS 01635000
  ELSE
    READFCN;
  START
  " >> Class C Material. Proprietary to Unisys Corporation <<"; 01636000
                                         01637000
                                         01638000010020002
                                         01638200010020002
                                         01638400010020002
                                         01639000
                                         01640000
                                         01641000

```

```

SHOWIT(TRUE);                                01642000
SHOWIT(FALSE);                               01643000
01644000
01645000
01646000
01647000
01648000
01649000
01650000
01651000
01652000
01653000
01654000
01655000
01656000
01657000
01658000
01659000
01660000
01661000
01662000
01663000
01664000
01665000
01666000
01667000
01668000
01669000
01670000
01671000
01672000
01673000
01674000
01675000
01676000
01677000
01678000
01679000
01680000
01681000
01682000
01683000
01684000
01685000
01686000
01687000
01688000
01689000
01690000
01691000
01692000
01693000
0169400010020003
0169420010020003
0169440010020003
0169460010020004
0169480010020004
01695000
01696000
01697000
01698000
016990001.005.000
01700000
01701000
01702000
01703000
0170400010020003
01705000
01706000
01707000
01708000

```

```

END
ELSE IF PL = "XM" THEN      * Express Mode - Limit displays      01709000010020003
  BEGIN
    EXPRESSMODE := TRUE;
    PL := * + 2;
    LL := * - 2;
  END;
  IF PL = "C" THEN FUNCTION := LOADCTLV      ELSE      01710000010020003
  IF PL = "D" THEN FUNCTION := LOADEVV      ELSE      01710100010020003
  IF PL = "V" THEN FUNCTION := VERIFYV      ELSE      01710200010020003
  SHOW("Unrecognized function selector",TRUE);
  END; * LL > 0
END; * WHILE

END; * SELECTOPTION
$$ PAGE

#####
# Epilog Procedure
#####
EPILOG PROCEDURE CLEANUP;
-----
# Purpose:          01724000
# Clean up any hanging threads.          01725000
# Parameters:       01726000
# Epilog procedures cannot have parameters. 01727000
# Globals:          01728000
# CTLATTACHED True: We have to un-attach it. 01729000
#-----          01730000
BEGIN
  REAL RSLT;
  IF CTLASSIGNED THEN          01731000
  BEGIN
    SHOW("Epilog procedure invoked to detach from CTL " C
          CTLUNIT FOR * DIGITS,TRUE);
    RSLT := USERMAINTREQUEST(CTLUNIT,UNASSIGNCTLV,0,0,0,
                           SHORTBUF,MRD,HDPRESULT);
    IF RSLT > 0 THEN          01732000
    BEGIN
      SHOW("Unable to detach from CTL " C
            CTLUNIT FOR * DIGITS      C
            ". Error "              C
            RSLT FOR * DIGITS,TRUE);
    END;
  END; * Control attached          01733000
  IF SDMASSIGNED THEN          01734000
  BEGIN
    SHOW("Epilog procedure invoked to detach from drive " C
          CTLUNIT FOR * DIGITS,TRUE);
    RSLT := USERMAINTREQUEST(CTLUNIT,UNASSIGNUNITV,0,0,0,
                           SHORTBUF,MRD,HDPRESULT);
    IF RSLT > 0 THEN          01735000
    BEGIN
      SHOW("Unable to detach from drive " C
            CTLUNIT FOR * DIGITS      C
            ". Error "              C
            RSLT FOR * DIGITS,TRUE);
    END;
  END; * Disk attached          01736000
END; * Epilog procedure
$$ PAGE

#####
# Outer Block
#####

```

```

INITIALIZE;                                01776000
MAINLOOP:
  WHILE TRUE DO
    BEGIN
      SELECTOPTION;
      CASE FUNCTION OF
        BEGIN
          LOADCTLV:
            OPTODO := UNASSIGNCTLV;      % CTL  PARAMETER FOR UMR
            VERIFYFILE;
            LOADCTLFW;
          LOADEVV:
            OPTODO := UNASSIGNUNITV;      % DRIVE PARAMETER FOR UMR
            VERIFYFILE;
            LOADEVFV;
          VERIFYV:
            VERIFYONLY := TRUE;
            VERIFYFILE;
            VERIFYONLY := FALSE;
          END;  % CASE FUNCTION OF
        END;  % WHILE TRUE
      GRANDXIT:
        SHOW ("<< DFAST Program is ENDING >>", TRUE);
        IF CODE.OPEN THEN
          CLOSE(CODE);
        IF NOT ODTMODE THEN
          CLOSE(RMT);
          CLOSE(LINE);
      END.

```